Routing in Multi-class Queueing Networks

Christopher Kirkbride

Supervisor : Dr Phil Ansell

A Thesis presented for the scientific degree of Doctor of Philosophy

School of Mathematics and Statistics
Faculty of Science, Agriculture and Engineering
University of Newcastle
Newcastle upon Tyne
NE1 7RU
United Kingdom

October 2004

NEWCASTLE UNIVERSITY LIBRARY
204 06154 7
Thesis L7833

Abstract

We consider the problem of routing (incorporating local scheduling) in a distributed network. Dedicated jobs arrive directly at their specified station for processing. The choice of station for generic jobs is open. Each job class has an associated holding cost rate. We aim to develop routing policies to minimise the long-run average holding cost rate.

We first consider the class of static policies. Dacre, Glazebrook and Niño-Mora (1999) developed an approach to the formulation of static routing policies, in which the work at each station is scheduled optimally, using the achievable region approach. The achievable region approach attempts to solve stochastic optimisation problems by characterising the space of all possible performances and optimising the performance objective over this space. Optimal local scheduling takes the form of a priority policy. Such static routing policies distribute the generic traffic to the stations via a simple Bernoulli routing mechanism. We provide an overview of the achievements made in following this approach to static routing. In the course of this discussion we expand upon the study of Becker *et al.* (2000) in which they considered routing to a collection of stations specialised in processing certain job classes and we consider how the composition of the available stations affects the system performance for this particular problem. We conclude our examination of static routing policies with an investigation into a network design problem in which the number of stations available for processing remains to be determined.

The second class of policies of interest is the class of dynamic policies. General DP theory asserts the existence of a deterministic, stationary and Markov optimal dynamic policy. However, a full DP solution may be unobtainable and theoretical difficulties posed by simple routing problems suggest that a closed form optimal policy may not be available. This motivates a requirement for good heuristic policies. We consider two approaches to the development of dynamic routing heuristics. We develop an idea proposed, in the context of simple single class systems, by Krishnan (1987) by applying a single policy improvement step to some given static policy. The resulting dynamic policy is shown to be of simple structure and easily computable. We include an investigation into the comparative performance of the dynamic policy with a number of competitor policies and of the performance of the heuristic as the number of stations in the network changes. In our second approach the generic traffic may only access processing when the station has been cleared of all (higher priority) jobs and can be considered as background work. We deploy a prescription of Whittle (1988) developed for RBPs to develop a suitable approach to station indexation. Taking an approximative approach to Whittle's proposal results in a very simple form of index policy for routing the generic traffic. We investigate the closeness to optimality of the index policy and compare the performance of both of the dynamic routing policies developed here.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Kevin Glazebrook, for all of the help and guidance given during my studies. Thanks also go to Dr. Phil Ansell for his additional supervision and vital introduction to programming. I would like to thank my friends and the staff and students of Newcastle University who made my time in Newcastle very memorable indeed. Special thanks go to my parents who provided tremendous support and encouragement throughout.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In our everyday lives we make demands for service or are asked to provide service ourselves. The nature of these service requirements are many and varied. On an individual level one resource that we can offer is our time and effort. Our time is a precious resource that we all try to allocate to a number of different projects with the aim of improving our current situation. The possible returns from such actions are entirely dependent upon the project. For example, by spending time in employment we improve our situation financially, by taking time to do the things we enjoy we increase our pleasure or sense of well-being. The actions we take are governed by what we would like to achieve.

The problem of efficient resource management is of great interest to many industries. These industries make decisions on how best to allocate their resources in order to meet certain goals. The strategies they adopt will depend upon the performance measures of interest. For example, in order to satisfy a fluctuating demand, electrical power companies have to control the output of their power stations. Financial institutions distribute their key resource, money, typically to maximise profit. The motivation for much of our work arises from the computing and communication industries. Computer and communication technologies play an increasingly important role in our lives and the development of these technologies has been rapid. Such advancements have brought about a number of problems that the scientific community need to be able to address. One main area of interest concerns the control and design of computer and communication systems. Kleinrock (2002) discussed the optimal design of data networks. Altman (2000) provided a helpful overview of contributions which relate to communications networks. However,

many of the problems encountered in the classical literature are too simple to address many contemporary applications. Consider the two following examples:

(i) *The Grid.* The pace of development within the computer industry in recent years has brought about wide-ranging benefits to users worldwide. High performance computing hardware is readily affordable and high speed network access, such as broadband, is now widely available. Coupled with the explosive growth of the World Wide Web these advancements provide excellent development opportunities for science and business communities. An area exciting considerable interest is a concept known as *"the Grid"*. In the computational Grid computing and storage resources are available as a commodity. A service provider makes various computational resources available to users via a collection of networked machines. These machines themselves may have other tasks to perform. Users submit requests (which may be of many different kinds) without knowing, or caring, where they will be processed. The problem is how to distribute these heterogeneous requests across the network so as to make the best possible use of system resources and to provide the best possible quality of service. The reader is referred to Foster and Kesselman (1998) for a full discussion of Grid environments. In order to meet the computational demands of a large diverse group of applications sophisticated computing environments will need to be employed. Braun *et al.* (2001) discussed high-performance heterogeneous computing (HC) environments, which are well suited to meet these requirements. They stated that 'one key feature in achieving the best possible performance from HC environments is the ability to assign effectively the applications to machines and schedule their execution'.

(ii) *Distributed Expertise.* Many businesses utilise call centres to operate their support and after-sales services. Becker *et al.* (2000) considered a problem motivated by call centres operated by companies producing a range of products or services. Customers will telephone such centres with requests for service or technical support. These requests are then routed to operators, preferably to an operator with the requisite

expertise in dealing with this type of request. If there is a high rate of requests regarding one particular service or product then the expert servers may become overloaded. To reduce the workload for these servers, and possibly even customer waiting times, some of the traffic will have to be directed to less expert servers. Becker *et al.* (2000) discussed how to do this routing optimally.

The above scenarios highlight the need for the study of problems in which many classes of customers seek service that is provided by servers of differing capabilities. In an environment within which service can be provided at one of several stations, a natural question which arises concerns how incoming requests might be assigned to those stations in a optimal manner. A modelling framework suited to the analysis of such *routing* problems is that of a queueing network. Here we present a general routing problem in which we are able to incorporate a multi-class job population and a collection of heterogeneous service stations which, as suggested by the motivating examples above, are a key areas of interest in modern communication and computer systems. Our interest lies in the development of *routing policies* (i.e. policies that govern the distribution of the service requests within the network) that seek to achieve some optimum level of performance for the network system.

1.1 The General Routing Problem

Multi-class routing problems represent a formidable challenge to analysis. For an overview of some of the practical issues involved in developing routing policies the reader is referred to Gelenbe and Pekergin (1993). Even in the context of very simple models, for example routing a single job class to a collection of homogeneous stations, such routing problems have proved difficult to analyse. See the introduction to Liu and Righter (1998) for a helpful overview of the theoretical literature. In Figure 1.1 we provide a simple pictorial representation of a general model for the routing problems under consideration. Our

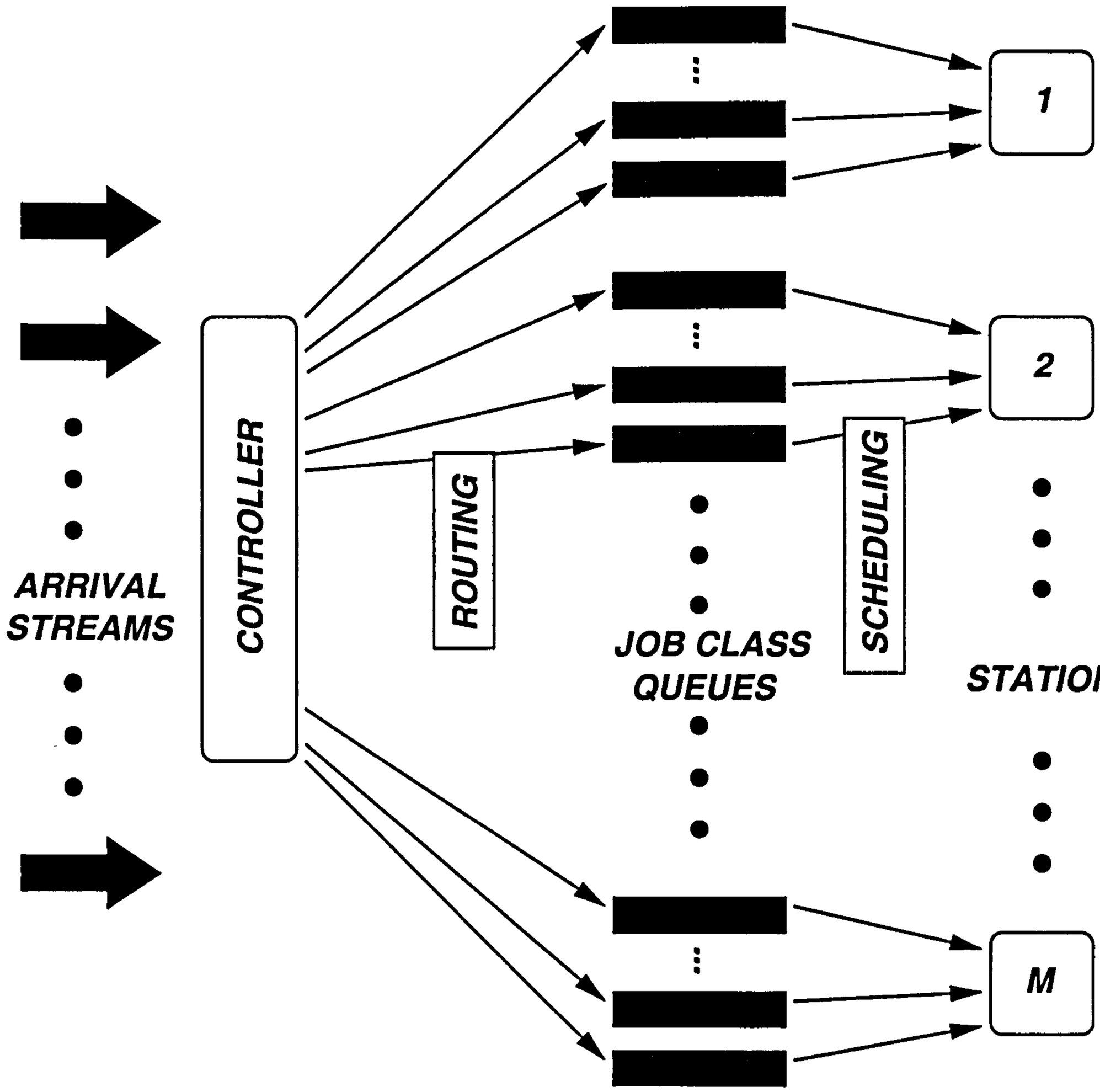

Figure 1.1: The general routing problem

service environment concerns a distributed service system comprising a system controller and a set of stations $\mathcal{M} = \{1, 2, \ldots, M\}$. Jobs (i.e. requests for service) from different customer classes arrive into the system at the system controller who will then assign these jobs to the stations within the network for processing. Once assigned to a station within the network the job will either begin service immediately, if processing capability is available, or join a queue at that station and wait until it is selected for processing. Each station operates a *local scheduling policy*. This policy determines how the queueing jobs will be selected for service. For example, one scheduling strategy is to process the

jobs in the order of their arrival at the station, this is commonly know as a *First Come First Served* (FCFS) policy. In their work on distributed computer systems Ross and Yao (1991) pointed to important features that may be present in such routing problems. They considered two broad types of arriving jobs, of which there may be several classes of each type. Some arriving jobs may be *dedicated* to a particular station and must be processed there. Other jobs classes are *generic* in that the choice of processing station remains open. A major innovation of Ross and Yao (1991) was that they supposed that each station schedules their work optimally as opposed to handling the work in a simpler FCFS fashion. They showed that considerable savings can be made when optimal local scheduling is incorporated into the routing problem. Under this model description the full routing problem comprises two key components, namely the routing policy applied by the system controller and the local scheduling rules adopted at each station. The development of the routing policy will have to take into account the local scheduling policy used.

A routing policy is naturally dependent upon the level of information available to the system controller and the sophistication of the network system. With increased information, such as full information on the system state (i.e. queue lengths at each station), the system controller will be able to implement policies that improve upon the performance of policies developed without access to this knowledge. In some systems it may not be possible to introduce the communication mechanisms required to store and manage the extra information. Additionally, the communication overheads may be too large as to be practical to accommodate them into the system. In choosing the routing policy there must be a balance between the value (or accessibility) of increased information and the requirements of the network system.

In the next two sections we briefly describe approaches to the development of routing policies on the basis of the level of information available to the system controller. We consider two classes of routing policies. The first class of policies are limited to utilising only the known system parameters and can make no use of information regarding the composition and length of the queues at each station. This class of state independent

policies are known as *static* routing policies. The second class of policies are state dependent in that they are able to incorporate knowledge of the composition and length of the queues at each station. This class of routing policies are known as *dynamic* routing policies.

1.2 Static Routing

In static routing problems the system controller can only base decisions regarding the assignment of jobs to service stations on the defining parameters of the system and can make no use of any knowledge of the present or past states of the system. For simple cases involving only a single job class and homogeneous stations round robin policies and Bernoulli routing with equal probabilities have been shown to be optimal. See, for example, Chang (1992), Ephremides *et al.* (1980), Koole (1996) and Liu and Towsley (1994). Note that round robin policies make no use of state information but do require the history of previous routing decisions to be known. In developing static routing policies for our multi-class problem we only consider policies based on time-independent information only. These static policies distribute the arriving jobs to the stations according to a fixed probability (Bernoulli routing).

For routing problems involving a single job class the question of how the traffic should be scheduled at each station is of limited concern, standard procedure is to schedule the work in a FCFS fashion. In the complex multi-class routing problems of interest here, the full routing problem is in fact a joint routing/local scheduling problem. To be able determine the routing probabilities for the job classes the local scheduling rules need to be taken into account.

The local scheduling sub-problem is itself a challenging stochastic optimisation problem. Many recent developments in the control of multi-class queueing systems have followed an approach to the optimisation problem using an approximation of the original system or some limiting process. Atkins and Chen (1995) considered a fluid model while

Harrison and Wein (1989) considered a diffusion process in heavy traffic. A control is then found that optimises the approximating systems. However, the subsequent evaluation of the controls for the original queueing system of interest can be very difficult.

An alternative approach to stochastic scheduling problems is the so-called *achievable region approach* (also known as the mathematical programming approach). In general terms the achievable region approach seeks solutions to stochastic optimisation problems by:

- Characterising the space of all possible performances of the stochastic system of interest.
- Solving a mathematical program over this region.

The space of all possible performances (the achievable region) is a polyhedron of special structure resulting in a mathematical program for which efficient algorithms exist. The optimising vector of performances for the mathematical program makes the identification of a control for the original system achieving this performance relatively straightforward. Early contributions to this approach were due to Coffman and Mitrani (1980) and Gelenbe and Mitrani (1980) who analysed multi-class $M/M/1$ and $M/G/1$ queueing systems. The approach was further developed by Shanthikumar and Yao (1992) and Bertsimas and Niño-Mora (1996). It has been shown for a number of important queueing models that the optimal local schedule takes the form of a *priority-index* policy. At a station each job class has an index associated with it, at each service decision epoch the server must select the job with the currently largest index.

Dacre (1999), Dacre and Glazebrook (2002) and Dacre, Glazebrook and Niño-Mora (1999) have developed an approach to static routing problems that makes extensive use of the achievable region approach. Their work encompasses a range of modelling possibilities for the controlled stochastic systems which are the individual stations in the network. We provide a detailed overview of the achievable region approach and its application in routing problems in Chapter 2 and we shall defer further elaboration of this approach until then.

1.3 Dynamic Routing

In dynamic routing problems the system controller has greater access to information than in the static models. The system controller has access to the composition and lengths of the queues at each station. A dynamic policy can, in principle, make decisions based upon the entire history of the system to date. Studies of queueing networks comprising homogeneous stations and a single job class have shown a "join the shortest queue" policy to be optimal for a number of system models. See, for example, Hordijk and Koole (1990), Weber (1978) and Winston (1977). A useful tool in the analysis and control of queueing systems in a dynamic framework is the *Markov decision process.*

1.3.1 Markov Decision Processes and Dynamic Programming

A Markov decision process is a model of a dynamic system evolving over time in which the progression through the system states is governed by a succession of transition probability distributions which in turn is controlled by the decision taken in the current state. In these models we assume that the Markov property holds. The Markov assumption states here that the transition to the next state depends only upon the effect of the action taken in the present system state and is not influenced by the past history of the system.

A *discrete-time* Markov decision process (MDP) is a dynamic system with decision epochs at equidistant points of time, $t = 0, 1, \ldots$. We have a set of possible system states I. For each state $i \in I$ a set of actions $A(i)$ are available. Depending upon the action a taken in state i an immediate cost $c_i(a)$ is incurred. At the next decision epoch the system will be in state j with probability $p_{ij}(a)$ where $\sum_{j \in I} p_{ij}(a) = 1,\ i \in I$.

MDPs are a powerful tool for the analysis of stochastic optimisation problems. Many of its applications are found in the optimal control of queueing systems. Due to the stochastic nature of the arrival and service processes in many queueing systems these problems are more naturally modelled by allowing decision epochs to occur randomly in time. A continuous time version of an MDP, known as a *semi*-Markov decision process

(SMDP), allows for time intervals between decision epochs to be random variables. The reader is referred to Puterman (1994) for a full discussion of MDPs/SMDPs and their applications.

The techniques of *Dynamic Programming* (DP) are a natural framework to finding optimal solutions to MDPs. The concept of DP was developed by Bellman (1957) as a computational approach to solving sequential decision problems. His principle of optimality states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

In applying Bellman's principle to the MDP problems above we obtain a set of recursive equations, the solution to these recursions result in the optimal cost function. If an exact analytical solution to the recursive equations is available then it may be possible to deduce structural properties of the optimal cost function from which an optimal policy can be deduced. When a full analytical solution is unavailable numerical methods can be applied to solve the recursions. However, numerical approaches are severely limited in their application to high dimensional problems and/or with a large number of states. Additionally, tackling such problems from a purely computational approach is generally lacking in any insight as to the structure of the optimal policies.

1.3.2 Restless Bandits

Restless Bandit Problems (RBPs) involve the sequential allocation of system resources among a collection of projects or bandits, which evolve stochastically over time. The aim being, say, to maximise the total expected rewards. The RBP, introduced by Whittle (1988), is a development of the classic *Multi-Armed Bandit* (MAB) problem of Gittins (1979). In Gittins' MAB problem service may be allocated to only one of the competing projects. The project to which service is allocated is said to be *active*. The remaining projects are *passive*. On selection of a project a reward is earned and the active project

then evolves according to a Markov transition law. All passive projects states remain unchanged. The optimal policy is an index policy known as the Gittins index policy. The Gittins index is a function of the project type and state. At each decision epoch the optimal policy is to select the project of currently largest index. In Whittle's (1988) RBP service may be allocated to more than one of the competing projects. Additionally, all projects continue to evolve whether they are active or passive according to active/passive transition laws. This active/passive evolution of the projects in RBPs make it a natural vehicle for a queueing system model. For an RBP comprising n projects, *exactly* $m < n$ of the projects must be active at all times. Whittle considered a relaxed version of the restless bandit problem in which m projects must be active *on average*. He applied a Lagrangian approach to the solution of the relaxed RBP. The Lagrange multiplier has the economic interpretation of a *subsidy for passivity*, the value of which ensures that m projects are active on average. He defines an index for each project in a given state as the value of subsidy that makes both active and passive actions optimal in that state. This index reduces to the Gittins index for the case $m = 1$ and the passive projects remain static. Whittle (1988) proposed an index policy for the RBP utilising the project indices above: at each decision epoch make active the m projects of currently largest index. This ensures the requirement that exactly m projects are active at all times is satisfied. However, Whittle's proposed index may not exist. The project must be shown to have an *indexability* property, which itself is difficult to establish. Even when the indices exist they are not guaranteed to be optimal.

1.4 Thesis Structure

In this section we give an outline structure of the remaining chapters of the thesis. Our work concerns routing in multi-class queueing networks and the development of routing policies for such problems. In the general routing model under consideration we have jobs from a number of classes arriving at a network system comprising a collection of service

stations offering (possibly) different service capabilities. The job classes themselves are of two distinct types. Dedicated jobs may only be processed at a specific station within the network and are assumed to arrive directly at that station. Generic jobs can be processed anywhere within the network. They arrive at the system controller who then assigns the jobs to a station. Each station schedules their work optimally. Our goal is the development of policies for routing the generic traffic across the network to minimise the long-run average holding cost rate. We consider two classes of routing policy for these problems. In Chapter 2 we discuss static (i.e. state independent) routing polices. Chapters 3 and 4 are concerned with the development of dynamic (i.e. state dependent) policies.

For the static routing problems considered in Chapter 2 the static routing policies take the form of a simple Bernoulli routing policy in which arriving jobs are assigned to a station within the network with some fixed probability. Our approach to the development of such static routing policies follows the achievable region approach as discussed by Dacre, Glazebrook and Niño-Mora (1999). We provide a detailed overview of this approach and its application to routing problems. We present four examples of station model to which the methodologies described in this chapter can be applied: the Klimov network, the $M/M/1$ queueing system, the $M/G/1$ queueing system and the $M/M/\eta$ queueing system with identical servers. These examples can be thought of as important representative cases. A key development in the achievable region approach by Bertsimas and Niño-Mora (1996) showed that systems satisfying *Generalised Conservation Laws* (GCL) have an achievable region that is an extended polymatroid. An adaptive greedy algorithm is available that optimally solves LPs over extended polymatroids. They classified the extreme points of the extended polymatroid and defined allocation indices as the sum of variables representing an optimal solution to the corresponding dual LP. A closed form expression is available for the optimal objective value of the LP. The allocation indices for extended polymatroids correspond with an index policy for GCL systems. A number of systems satisfying GCL have the important structural properties of *decomposability* and

reducibility. The notion of decomposability was introduced by Bertsimas and Niño-Mora (1996). Systems that exhibit this strong property can be solved by breaking them down into a number of smaller sub-problems. Reducibility, introduced by Garbe and Glazebrook (1998), guarantees that systems formed by restricting service to different subsets of jobs are related to each other in a fairly simple way. Dacre and Glazebrook (2002) introduced the more general notion of *quasi-reducibility.* They showed that GCL systems that are decomposable and quasi-reducible with respect to a partition of job classes exhibit a form of supermodularity in the optimal return. Dacre (1999) and Dacre and Glazebrook (2002) used these supermodularity results to deduce properties of the optimal cost function as a function of job arrival rates. They showed that these cost functions exhibit a form of partial convexity in the job arrival rates. In systems satisfying these properties a solution to the full routing problem exists. In special cases the partial convexity property translates into full convexity, this in turn yields full convexity in the full routing problem. For such problems efficient numerical algorithms exist that will provide optimal solutions. In problems of routing to homogeneous stations, *Equal Splitting Policies* (ESP) are shown to be optimal when the station returns are convex. However, this is not the case when the station returns are partially convex. We discuss optimisation approaches that can be applied in the determination of the optimal routing probabilities.

The final sections apply the techniques developed in formulating optimal static routing policies. We consider the routing problem investigated by Becker *et al.* (2000). Calls (jobs) of different types are to be routed to operators (service stations) within a call centre. Ideally calls should be handled by operators who specialise in dealing with these types of call, however, this is not always possible and some calls may have to be routed to less expert operators. Becker *et al.* discussed how to do this routing optimally for problems in which each service station schedules their traffic according to a FCFS policy and an optimal scheduling policy. We extend their computational study to consider a greater range of system set-ups. In all problems operating an optimal local scheduling policy at each station, in comparison to FCFS scheduling, reduces call delays. For these problems

a specialist operator is available for each type of call. Under limited budget constraints this may not be possible. Our optimisation problem now concerns the determination of the optimal mix of specialist stations to be built into the network. We carry out a numerical investigation into this problem and suggest a reasonable heuristic policy. We conclude the chapter with an investigation of a problem closely related to our routing problems, network design. Network design problems concern the construction of a network system that will meet the service demands placed upon it. We consider constructing networks of homogeneous stations. We show that under ESP routing and where the station returns are partially convex additional stations contribute diminishing returns. We conduct a numerical investigation into a simple network design problem. We consider the performance of networks constructed under ESP routing for a range of system parameters.

In Chapter 3 we describe an approach the development of dynamic routing policies for multi-class systems. The work developing this approach and the resulting dynamic policies has been published: see Ansell, Glazebrook and Kirkbride (2003). In the development of these dynamic policies we apply a single policy improvement step to an optimal static policy. In this we follow an idea proposed by Krishnan (1987) and discussed by Tijms (1994) in the context of simple single class systems. We consider problems in which each station is modelled as a multi-class $M/M/1$ queueing system. The resulting dynamic policies are shown to have a simple and intuitive structure that generalise "join the shortest queue" policies to our complex multi-class structure in a natural way and are based upon a index for each station in a given state. These indices are a measure of the congestion at each station and are shown to be linear in the class specific queue lengths there. Hence, in its application the dynamic routing policy will route an incoming generic job to the station with the smallest current index value. The coefficients of this linear measure are shown to reflect both the individual station dynamics and the class of arriving job. Further simplifications arise in special cases. The analysis is extended to consider a problem in which the individual stations are modelled as Klimov networks. We report a computational investigation into the performance of these dynamic routing policies with

a number of competitor policies. These include the optimal static routing policy, round robin, JSQ and the optimal dynamic routing policy. In the final section we consider utilising these dynamic routing policies in the network design problem introduced in Chapter 2. In a computational investigation we consider the benefits to a system architect in constructing systems that incorporate these dynamic policies over systems constructed under the ESP routing.

In Chapter 4 we develop an alternative approach to the development of dynamic routing policies. The work developing this approach and its resulting dynamic policies has also been published: see Glazebrook and Kirkbride (2004). We utilise ideas related to the class of restless bandit problems in the development of an index policy for our routing problem. We consider problems in which the individual stations are modelled as two-class $M/M/1$ queueing systems. We take an approximative approach to Whittle's (1988) proposal for indexability to develop a station index. In this we consider a class of admission control problems involving a single station and develop the station index via the application of a single policy improvement step to an optimal static policy for this single station problem. The indices for the station are shown to be increasing and non-linear in the station workload. The policy implemented will route an incoming generic job to the station of currently smallest index. We extend the analysis to a problem in which the stations are modelled as Klimov networks. We conclude with a numerical study investigating the performance of the index policy with the optimal static routing policy, the dynamic routing policy developed in Chapter 3 and the optimal dynamic routing policy. The numerical evaluations of the cost rates for the dynamic policies in Chapters 3 and 4 are performed via DP value iteration (see, Tijms (1994)).

Chapter 2

Static Routing

2.1 Introduction

The general routing problem discussed in Chapter 1 concerns the distribution of work in a diversified service system. Such *routing* problems represent a significant challenge to analysis. Here our attention focuses on the static routing problem. For such problems the system controller (whose task is to distribute the workload across the network) only has access to information regarding the defining system parameters with which to guide routing decisions. The policy implemented will route an arriving job to a station according to a fixed probability. Our interest lies in the development of optimal static routing policies for this problem. The study of routing problems in these contexts is important. For example, in network systems in which it is not possible to introduce mechanisms to store and manage the required state information for the implementation of dynamic policies. This may be due to the sophistication of the network system or simply that the costs from implementing these mechanisms are too large.

In this chapter we introduce the static routing problem. These problems allow for multiple job classes of different types and stations with varying service capabilities. A feature of these models is that we suppose the workload at the individual stations is scheduled optimally. The full routing problem is shown to be a joint routing/local scheduling problem. In determining the optimal local scheduling policy we make extensive use of the achievable region approach. We provide a detailed overview of the achievements made in following this approach and its application in the development of static routing policies

for a range of system models. The performance of these policies are then numerically assessed in a problem which considers routing jobs to a network of stations where each station possesses varying degrees of specialisation in processing the different job classes. The chapter ends with consideration of the related problem of network design.

2.2 The Static Routing Problem

A distributed system comprises a set of stations $\mathcal{M} = \{1, 2, \ldots, M\}$ and a system controller. Jobs from a number of different classes arrive at the system for processing. Jobs (and the classes to which they belong) are either *generic* or *dedicated*. Generic jobs arrive at the system controller who will immediately route the arrival to one of the stations within the network. Generic jobs can be processed at any station. Dedicated jobs can be processed only by a specified station and are assumed to arrive directly at that station. We denote the set of classes of generic jobs by G and the set of classes of jobs dedicated to station m by D_m, $m \in \mathcal{M}$. Hence $E = G \cup D_1 \cup \ldots \cup D_M$ is the set of job classes allowed access to the system while $E_m = G \cup D_m$ is the set of job classes allowed access to station $m \in \mathcal{M}$.

Jobs arrive at the system as independent Poisson streams, having rates (mean number of arrivals per unit time) λ_j, $j \in E$. Let $\boldsymbol{\lambda} \equiv \{\lambda_j\}_{j \in E}$ and $\boldsymbol{\lambda}^G \equiv \{\lambda_g\}_{g \in G}$ denote the vectors of arrival rates. To conform with other notation we shall write $\lambda_j = \lambda_{jm}$, $j \in D_m$, $m \in \mathcal{M}$ for the arrival rates of the dedicated classes. Here we only consider the class of *static* routing policies. Hence, when a generic job class of g arrives, the system controller assigns it to station m with fixed probability p_{gm}. The *controller routing matrix* $\mathbf{P} = (p_{gm})_{g \in G, m \in \mathcal{M}}$ satisfies $\mathbf{Pe} = \mathbf{e}$ where $\mathbf{e}$ is a $|G|$-vector of 1's, so an arriving job is routed to a station with probability 1. We use $\boldsymbol{\lambda}_m^G = \{p_{gm}\lambda_g^G\}_{g \in G}$ to denote the vector of generic arrival rates to station $m \in \mathcal{M}$.

Under this scheme, jobs from classes in $G \cup D_m$ arrive at station m and will compete for processing capacity with those already there. Hence, each station operates a *local*

scheduling policy, namely a rule for determining which job class to process at each decision epoch. Let $\mathcal{U}_m$ be the set of *admissible scheduling policies* for station m. An admissible policy, $u_m \in \mathcal{U}_m$, must satisfy the natural restrictions that it is *non-anticipative* (i.e. decisions are only based on the past and present state of the system) and *work conserving* (i.e. servers are never idle when there are jobs in the system and will not allow jobs to leave until their processing requirements have been met).

A holding cost, $c_j \geq 0$, is associated with each job class $j \in E$. Our goal is to route the generic workload across the stations and schedule the work at each station to minimise the long-run average holding cost rate:

$$\sum_{m \in \mathcal{M}} \sum_{j \in G \cup D_m} c_j E(N_{jm}) \tag{2.1}$$

where N_{jm} is the number of class j jobs at station m and the expectation is taken in steady state. Note that we assume that the system is *stable*, in that a steady state solution with finite queue lengths exists. We shall assume that all arguments of the optimal cost functions Φ and Ψ, which we now define, result in a stable system.

If we fix the generic arrival rates, $\boldsymbol{\lambda}_m^G$, at station m then the local scheduling problem becomes

$$\Psi_m(\boldsymbol{\lambda}_m^G) = \inf_{u_m \in \mathcal{U}_m} \sum_{j \in G \cup D_m} c_j E_{u_m}(N_{jm}). \tag{2.2}$$

The routing aspect of the minimisation problem in (2.1) can now be expressed as

$$\Phi_{\mathcal{M}}(\boldsymbol{\lambda}^G) = \min_{\mathbf{P}} \sum_{m \in \mathcal{M}} \Psi_m(\boldsymbol{\lambda}_m^G). \tag{2.3}$$

Plainly, an ability to compute and/or characterise the optimal returns Ψ_m as functions of the generic load $\boldsymbol{\lambda}_m^G$ is important for the solution of the load balancing problem in (2.3). We now develop a range of models for local scheduling at station m all of which enable us to say a great deal about the Ψ_m, $m \in \mathcal{M}$. The models presented are not an exhaustive

selection but they can be considered to be a number of important representative cases to which we can apply the methodology presented in this chapter. We could, for example, model station m as an undiscounted branching bandit (see Dacre and Glazebrook (2002)).

Example 1 (Station m a Klimov network)

Jobs arrive at station m in independent Poisson streams and have service requirements which are exponentially distributed. Arrival rates and service rates for class j are λ_{jm} and μ_{jm} respectively, $j \in G \cup D_m$. Once a job from class i has completed service, it may be routed for further service as a class j job, with probability $q_{ij}^m, i, j \in G \cup D_m$, or it may leave the system, with probability $q_{i0}^m = 1 - \sum_{j \in G \cup D_m} q_{ij}^m$. Here it will usually be appropriate to assume that generic jobs or dedicated jobs are routed for further service as generic or dedicated jobs respectively. To ensure that a job entering the station leaves it with probability one we require that the matrix $\mathbf{I}^m - \mathbf{Q}^m$ is invertible, where $\mathbf{Q}^m = \{q_{ij}^m\}_{i,j \in G \cup D_m}$ and $\mathbf{I}^m$ is the $|G \cup D_m| \times |G \cup D_m|$ identity. We further assume that all arrival, service and routing processes are mutually independent. A single machine provides service. The admissible scheduling controls, $\mathcal{U}_m$, at station m are non-anticipative and work conserving. They will select a single job for service (provided the station is non-empty) at each arrival and service completion epoch. Under this model, the traffic at a station can be quite general in structure. We can, for example, provide for jobs which have a (current) state which evolves under processing as a continuous time Markov process.

Example 2 (Station m a multi-class $M/M/1$ queue)

Jobs arrive at station m in independent Poisson streams and have service requirements which are exponentially distributed. Arrival rates and service rates for class j are λ_{jm} and μ_{jm} respectively, $j \in G \cup D_m$. Once jobs are served, they leave the system. Arrival and service processes are mutually independent. A single machine provides service. The admissible scheduling controls, $\mathcal{U}_m$, at station m are non-anticipative and work conserving. They will select a single job for service (provided the station is non-empty) at each arrival

and service completion epoch. It is easy to see that this station model is equivalent to the no-feedback version of the Klimov network in Example 1 (i.e. $q_{ij}^m = 0,\ j \neq 0$).

Example 3 (Station m a multi-class $M/G/1$ queue)

Jobs arrive at station m in independent Poisson streams and have general service requirements which are i.i.d. within each class. The class j arrival rate is λ_{jm} and class j service times have mean μ_{jm}^{-1} and finite second moment $M_{jm},\ j \in G \cup D_m$. Once jobs are served, they leave the system. Arrival and service processes are mutually independent. A single machine provides service. The admissible scheduling controls, $\mathcal{U}_m$, at station m are non-anticipative and work conserving with the further restriction that controls should be non-preemptive (i.e. once service has started on a job it will be carried through to completion). They will select a single job for service (provided the station is non-empty) at each service completion epoch.

Example 4 (Station m a multi-class $M/M/\eta_m$ queue with identical service rates)

Jobs arrive at station m in independent Poisson streams and have service requirements which are exponentially distributed. Arrival rates and service rates for class j are λ_{jm} and μ_m respectively, $j \in G \cup D_m$. Once jobs are served, they leave the system. Arrival and service processes are mutually independent. Service is provided by η_m machines working in parallel and any machine can process any job. The admissible scheduling controls, $\mathcal{U}_m$, at station m are non-anticipative and work conserving. They will select a collection of jobs for service at each arrival and service completion epoch. If there are η_m or fewer jobs present at such an epoch then all jobs will be chosen while if there are more than η_m jobs present then a collection of η_m of them will be chosen, one to be processed on each machine.

2.3 The Achievable Region Approach

Many stochastic control problems, including those involving the multi-class queueing systems of the previous section, have been successfully analysed via the so-called achievable region approach (see Dacre, Glazebrook and Niño-Mora (1999)). The advantage in following this approach is that it remains in close contact to the original stochastic system as opposed to approaches which study fluid or diffusion approximations to the original system. Typically, such analyses allow us to make strong statements about the control policies identified. The achievable region approach to the optimal control of stochastic systems (alternatively known as the mathematical programming approach) attempts to solve stochastic optimisation problems by:

(i) identifying the performance space $\mathcal{X}$, namely the set of all possible system performances (the achievable region) under admissible scheduling controls;

(ii) reformulating the stochastic optimisation problem of interest as a mathematical programming problem with feasible region $\mathcal{X}$ and;

(iii) identifying a control that corresponds to the optimal solution of the stochastic optimisation problem via a solution to the above mathematical programming problem.

Due to the nature of the current work we develop the method of the achievable region approach in the context of multi-class queueing systems appropriate to the analysis of the local scheduling problem given by (2.2). To avoid notational complexities we shall focus on a single station m in the network and drop the station suffix from the notation. We return the suffix at the end of Section 2.7 where we consider the full routing/local scheduling problem of (2.3).

Queues from classes in $G \cup D$, form at the station which operates an admissible scheduling policy $u \in \mathcal{U}$ which has an associated *system performance vector* $\mathbf{x}^u = (x_1^u, x_2^u, \ldots, x_{|G \cup D|}^u)$. Here x_j^u is the expectation of a performance measure related to class j. We shall typically take the performance measure x_j^u to be the long-run average number

of class j jobs at the station under control u, denoted $E_u(N_j)$. This seems a natural choice of performance measure for many queueing control problems. However, in certain problem instances it can be beneficial to consider others. The *performance space* is the set of possible performances, denoted $\mathcal{X} = \{\mathbf{x}^u : u \in \mathcal{U}\}$. Given the holding cost vector $\mathbf{c} = (c_1, c_2, \ldots, c_{|G \cup D|})^T$, the stochastic scheduling problem can be expressed as

$$\Psi = \inf_{u \in \mathcal{U}} (\mathbf{c}^T \mathbf{x}^u). \tag{2.4}$$

Our aim is to identify a control u^{OPT} that attains the infimum in (2.4). If $\mathcal{X}$ is known we can solve the minimisation problem

$$\Psi = \inf_{\mathbf{x} \in \mathcal{X}} (\mathbf{c}^T \mathbf{x}) \tag{2.5}$$

and then identify a control u^{OPT} that achieves $\mathbf{x}^{OPT}$, the optimising vector.

We can identify the exact performance space for the station models considered in Examples 1-4. The performance spaces have been shown to have a special structure that allows for the efficient solution of the mathematical program in (2.5). The form of the optimal solutions makes the identification of the optimal scheduling control relatively straightforward. The optimal scheduling control in every case is a *strict priority policy.* Such policies give priority to jobs in class i over jobs in class j, if at each decision epoch, the server will only select a class j job if there are no class i jobs present in the system. Given $\pi = (\pi_1, \pi_2, \ldots, \pi_{|G \cup D|})$ a permutation of $G \cup D$, priority policy π will give priority to the job classes according to permutation π with jobs in class π_1 having priority over all other job classes and jobs in class $\pi_{|G \cup D|}$ having the lowest priority.

Early contributions to the development of the achievable region approach were due to Coffman and Mitrani (1980) and Gelenbe and Mitrani (1980). Their ground-breaking work analysed stochastic scheduling problems concerning multi-class $M/M/1$ and $M/G/1$ queueing systems. We now briefly outline their approach on the simple problem of a two-class $M/M/1$ queueing system.

The two-class $M/M/1$ queueing system considered here is a version of the station model of Example 2 in which the are 2 generic job classes and no dedicated traffic. We denote the traffic intensity due to class j jobs by $\rho_j = \lambda_j/\mu_j,\ j = 1,2$. To guarantee stability, the total traffic intensity $\rho = \rho_1 + \rho_2$ is assumed to be less than 1. The goal is to find a minimising control $u \in \mathcal{U}$ for the long-run holding cost rate, i.e.

$$\inf_{u\in\mathcal{U}} \{c_1 E_u(N_1) + c_2 E_u(N_2)\}, \tag{2.6}$$

where c_j is a (linear) holding cost rate and N_j is the number of class j jobs at the station and E_u is an expectation taken with respect to the steady state distribution of the system under policy u.

By consideration of the work-in-system, denoted $V_u(t)$, whose value is equal to the sum of remaining service times of all customers in the system at time t we can identify a collection of equations and inequalities involving the steady state expectations $E_u(N_1)$ and $E_u(N_2)$. These are collectively known as the *conservation laws* defining the system (see Kleinrock (1976)). Under admissible scheduling policy u sample paths of $V_u(t)$ consist of upward jumps at job arrival epochs equal to the amount of additional processing requirement followed by a period of service during which the work-in-system is reduced at rate 1. Service periods terminate either when the system is empty ($V_u(t) = 0$) or with an upward jump indicating a job arrival. It is clear that the sample paths of $V_u(t)$ are independent of the choice of admissible u. It then follows that the steady state work-in-system is constant over all admissible scheduling polices. Under the assumption of exponential service times we may express the steady state expected work-in-system as $E_u(N_1)\mu_1^{-1} + E_u(N_2)\mu_2^{-1}$, solutions for which are well known under the first come first served (FCFS) scheduling policy. For our simple two-class system this equality constraint is given by

$$\frac{E_u(N_1)}{\mu_1} + \frac{E_u(N_2)}{\mu_2} = \frac{\rho_1\mu_1^{-1} + \rho_2\mu_2^{-1}}{1-\rho_1-\rho_2}, \quad u \in \mathcal{U}. \tag{2.7}$$

We now consider $V_u^1(t)$, the work-in-system at time t due to class 1 jobs only. This quantity is not control invariant due to its dependence upon the distribution of service between the two classes under admissible scheduling policy u. Sample paths of $V_u^1(t)$ may have horizontal segments when $V_u^1(t) > 0$ corresponding to periods in which class 1 jobs are present in the system but class 2 jobs are currently being processed. For each realisation of the system it is clear that $V_u^1(t)$ is minimised for each t by the policy u which always gives class 1 jobs priority over class 2. Hence the steady state expected class 1 work-in-system is minimised by this (preemptive) priority policy, denoted $1 \rightarrow 2$. We thus obtain the following inequality

$$\frac{E_u(N_1)}{\mu_1} \geq \frac{\rho_1 \mu_1^{-1}}{1-\rho_1}, \quad u \in \mathcal{U}, \tag{2.8}$$

where the r.h.s. of inequality (2.8) is the mean amount of work-in-system of an $M/M/1$ system serving class 1 jobs only. The same reasoning gives the corresponding result that

$$\frac{E_u(N_2)}{\mu_2} \geq \frac{\rho_2 \mu_2^{-1}}{1-\rho_2}, \quad u \in \mathcal{U}, \tag{2.9}$$

where the r.h.s. of inequality (2.9) is attained by (preemptive) priority policy $2 \rightarrow 1$. Without further discussion we state that these *conservation laws* define the performance space $\mathcal{X}$ exactly. The performance space is given by the region

$$\begin{aligned}\mathcal{X} &= \left\{ \left(\frac{E_u(N_1)}{\mu_1}, \frac{E_u(N_2)}{\mu_2} \right); u \in \mathcal{U} \right\} \\ &= \left\{ (x_1, x_2) \in \mathbb{R}_+^2 : x_1 \geq \frac{\rho_1 \mu_1^{-1}}{1-\rho_1}; x_2 \geq \frac{\rho_2 \mu_2^{-1}}{1-\rho_2}; x_1 + x_2 = \frac{\rho_1 \mu_1^{-1} + \rho_2 \mu_2^{-1}}{1-\rho_1-\rho_2} \right\}.\end{aligned}$$

Figure 2.1 illustrates the line segment defining the performance space.

Now consider the LP

$$\Psi = \inf_{\mathbf{x} \in \mathcal{X}} \{ c_1 \mu_1 x_1 + c_2 \mu_2 x_2 \}. \tag{2.10}$$

The minimum in (2.10) is attained at end-point A when $c_1\mu_1 \geq c_2\mu_2$ and at end-point B otherwise.

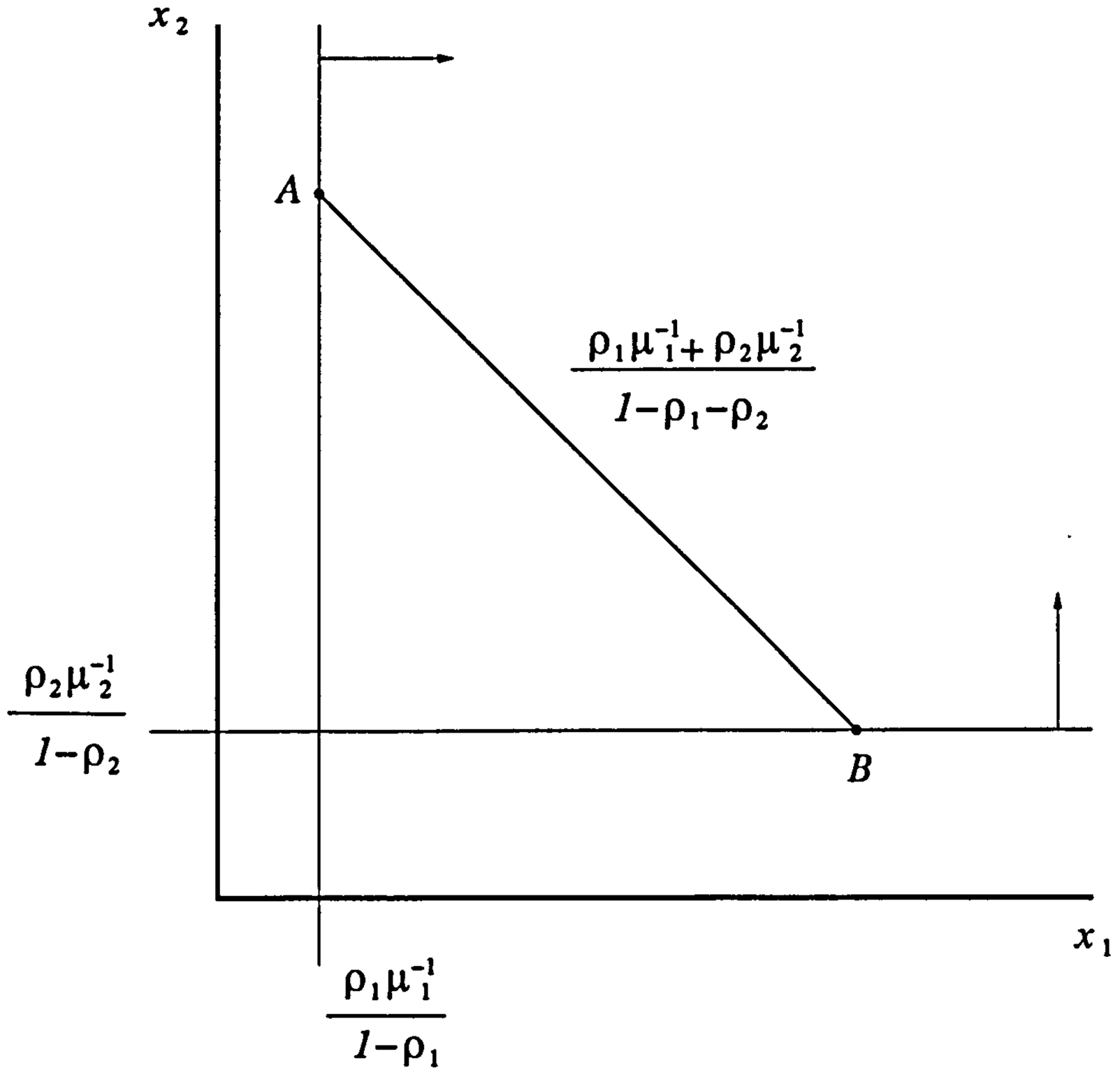

Figure 2.1: The line segment $\mathcal{X}$

We now return to the original objective, namely identifying the control u^{OPT} that solves (2.6), rewritten as

$$\Psi = \inf_{u\in\mathcal{U}} \{c_1\mu_1 x_1^u + c_2\mu_2 x_2^u\}, \tag{2.11}$$

where in (2.11) we take $x_j^u = E_u(N_j)/\mu_j$ as the class j performance under control u. When $c_1\mu_1 \geq c_2\mu_2$, $\mathbf{x}^{OPT} = A$ and the control achieving this is $1 \to 2$. When $c_2\mu_2 \geq c_1\mu_1$, $\mathbf{x}^{OPT} = B$ and this achieved by $2 \to 1$. We conclude that the control solving (2.11) is one in which priority is given to the job class with the larger $c_j\mu_j$ value. Hence the optimal control favours options which drive down the holding cost rate most rapidly. This so-called

$c\mu$-rule is a classic result from queueing control theory (see Cox and Smith (1961)).

2.4 Strong Conservation Laws

The achievable region approach was developed further by Shanthikumar and Yao (1992). They were able to show that for any performance vector satisfying *strong conservation laws* (SCL) the performance space is (the base of) a *polymatroid.* The solutions of LPs whose feasible space is a polymatroid are well known and will correspond to strict priority policies for our multi-class scheduling problems.

A performance vector is said to satisfy SCL if the total performance over all job classes $G \cup D$ is invariant under all admissible policies, and if its minimal performance over job classes in a subset $S \subseteq G \cup D$ is achieved by any policy that gives priority to job classes in S over job classes in S^c. More formally, we have Definition 1.

Definition 1 (Strong Conservation Laws)

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G \cup D|})$ *satisfies strong conservation laws (SCL) if there exists a set function* $b : 2^{G \cup D} \rightarrow \mathbb{R}_+$ *satisfying, for all* $u \in \mathcal{U}$,

$$\sum_{j \in G \cup D} x_j^u = b(G \cup D), \tag{2.12}$$

$$\sum_{j \in S} x_j^u \geq b(S), \text{ for all } S \subset G \cup D, \tag{2.13}$$

and such that for all priority policies $\pi = \{\pi_1, \pi_2, \ldots, \pi_{|G \cup D|}\}$,

$$\sum_{j=1}^{r} x_{\pi_j}^{\pi} = b(\{\pi_1, \pi_2, \ldots, \pi_r\}), \text{ for } r = 1, 2, \ldots, |G \cup D|. \tag{2.14}$$

We say the system satisfies SCL(b).

If the performance vector $\mathbf{x}$ satisfies SCL then the corresponding achievable region

$\mathcal{X} = \{\mathbf{x}^u; u \in \mathcal{U}\}$ is contained in the polyhedron $\mathcal{P}$ defined by

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}_+^{|G \cup D|} : \sum_{j \in S} x_j \geq b(S), S \subset G \cup D; \sum_{j \in G \cup D} x_j = b(G \cup D) \right\}. \tag{2.15}$$

If the set function $b : 2^{G \cup D} \rightarrow \mathbb{R}_+$ is normalised, non-decreasing and supermodular, namely,

$$\begin{aligned}
b(\emptyset) &= 0, && \text{(normalised)},\\
b(S) \leq b(T),\ & S \subseteq T \subseteq G \cup D, && \text{(non-decreasing)},\\
b(S \cup \{j\}) - b(S) &\leq b(T \cup \{j\}) - b(T), \\
S \subseteq T \subseteq G \cup D & \text{ and } j \notin T, && \text{(supermodular)},
\end{aligned}$$

then $\mathcal{P}$ defines the (base of a) (contra) *polymatroid* with base function b. Shanthikumar and Yao (1992) were able to show that strict priority policies guarantee the *supermodularity* of b, giving rise to the following result.

Theorem 1 (Shanthikumar and Yao (1992))

If a performance vector $\mathbf{x}$ *satisfies strong conservation laws (2.12)-(2.14) then*

(a) *The polyhedron* $\mathcal{P}$ *in (2.15) is the performance space;*

(b) $\mathcal{P}$ *is (the base of) a polymatroid;*

(c) *The vertices of* $\mathcal{P}$ *are the performance vectors of absolute priority rules.*

The vertices of the polyhedron correspond to the minimum of some linear objective which explains the optimality of strict priority policies. We shall see later that this result is a special case of the more general Theorem 6. Many multi-class queueing systems including Examples 2-4 satisfy SCL. However, this is not true of Example 1 which is itself solved by a priority policy. In the next section we shall introduce *generalised conservation laws* (GCL). The performance space of systems that satisfy GCL corresponds to a type of polyhedron known as (the base of) an extended (contra) polymatroid.

2.5 Extended Polymatroids and Generalised Conservation Laws

Extended polymatroids (and polymatroids) have a role in stochastic scheduling problems solved by priority index policies analogous to that of classical polymatroids in combinatorial optimisation problems solved by greedy algorithms. They were introduced by Tsoucas (1991) and developed by Bertsimas and Niño-Mora (1996).

Definition 2 (Extended Polymatroid)

Given a set $G \cup D = \{1, 2, \ldots, |G \cup D|\}$, *a non-negative set function* $b : 2^{G \cup D} \to \mathbb{R}_+$, *and a matrix* $\mathbf{V} = (V_j^S)_{j \in G \cup D,\ S \subseteq G \cup D}$ *satisfying* $V_j^S > 0$ *for* $j \in S,\ S \subseteq G \cup D$, *the polyhedron*

$$\mathcal{P}(\mathbf{V}, b) = \left\{ \mathbf{x} \in \mathbb{R}_+^{G \cup D} : \sum_{j \in S} V_j^S x_j \geq b(S), S \subset G \cup D; \sum_{j \in G \cup D} V_j^{G \cup D} x_j = b(G \cup D) \right\}$$

is (the base of) an extended (contra) polymatroid if $\mathbf{v}(\pi) \in \mathcal{P}(\mathbf{V}, b)$ *for every permutation* π *of* $G \cup D$, *where, for* $\pi = (\pi_1, \pi_2, \ldots, \pi_{|G \cup D|})$, $\mathbf{v}(\pi)$ *is the unique solution to the set of simultaneous linear equations*

$$\sum_{j=1}^{r} V_{\pi_j}^{\{\pi_1, \pi_2, \ldots, \pi_r\}} x_{\pi_j} = b(\{\pi_1, \pi_2, \ldots, \pi_r\}),\ r = 1, 2, \ldots, |G \cup D|.$$

Many multi-class scheduling problems can be formulated as linear programs over extended (contra) polymatroids. The strong structural properties of these linear programs allow for efficient solution. Consider the Linear Program:

$$\begin{aligned} \text{minimise} \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{P}(\mathbf{V}, b) \end{aligned} \tag{2.16}$$

where $\mathbf{c} \in \mathbb{R}_+^{|G \cup D|}$, given by $\mathbf{c} = (c_1, c_2, \ldots, c_{|G \cup D|})$ is a cost rate vector.

Tsoucas (1991) proved the optimality of an adaptive greedy algorithm for solving (2.16). The adaptive greedy algorithm takes the cost vector $\mathbf{c}$ and the matrix $\mathbf{V}$ as its inputs. The output of the algorithm includes a permutation π of $G \cup D$ and a dual solution of the linear program (2.16), $\bar{\mathbf{y}}$. The steps of the algorithm proceed as follows:

Input: $(\mathbf{c}, \mathbf{V})$

Step: $k = |G \cup D|$

$$\text{set } S_{|G\cup D|} = G \cup D;$$

$$\text{set } \bar{y}^{S_{|G\cup D|}} = \min_{i \in S_{|G\cup D|}} \left\{ \frac{c_i}{V_i^{S_{G\cup D}}} \right\};$$

$$\text{select } \pi_{|G\cup D|} \in \operatorname*{argmin}_{i \in S_{|G\cup D|}} \left\{ \frac{c_i}{V_i^{S_{G\cup D}}} \right\};$$

Step: $k = |G \cup D| - 1, \ldots, 1$

$$\text{set } S_k = S_{k+1} \backslash \{\pi_{k+1}\};$$

$$\text{set } \bar{y}^{S_k} = \min_{i \in S_k} \left\{ \frac{c_i - \sum_{j=k+1}^{|G\cup D|} V_i^{S_j} \bar{y}^{S_j}}{V_i^{S_k}} \right\};$$

$$\text{select } \pi_k \in \operatorname*{argmin}_{i \in S_k} \left\{ \frac{c_i - \sum_{j=k+1}^{|G\cup D|} V_i^{S_j} \bar{y}^{S_j}}{V_i^{S_k}} \right\};$$

Step: 0

$$\text{set } \bar{y}^S = 0 \text{ for } S \in 2^{G\cup D} \backslash \{S_{|G\cup D|}, S_{|G\cup D|-1}, \ldots, S_1\}.$$

Output: $(\bar{\mathbf{y}}, \pi)$

In each of the $|G \cup D|$ steps the adaptive greedy algorithm selects an element of $G \cup D$ that achieves the minimum of an expression, modified at each stage, so that at the termination of a single pass through the algorithm we are provided with a solution

to the linear program (2.16). The next result establishes the optimality of the adaptive greedy algorithm. Bertsimas and Niño-Mora (1996) developed a proof based on linear programming duality.

Theorem 2 (Optimality of the Adaptive Greedy Algorithm)

If $(\bar{\mathbf{y}}, \pi)$ *is the output from running the adaptive greedy algorithm on* $(\mathbf{c}, \mathbf{V})$, *then* $\mathbf{v}(\pi)$ *solves the LP (2.16) and* $\bar{\mathbf{y}}$ *solves its dual.*

The optimality of the greedy algorithm allows us to explicitly characterise the extreme points of the extended (contra) polymatroid $\mathcal{P}(\mathbf{V}, b)$.

Theorem 3 (Extreme Points of $\mathcal{P}(\mathbf{V}, \mathbf{b})$)

The set of extreme points of $\mathcal{P}(\mathbf{V}, b)$ *is*

$$\{\mathbf{v}(\pi) \ : \ \pi \text{ is a permutation of } G \cup D\}.$$

In the special case where $V_j^S = 1$ for all $j \in S$, the region $\mathcal{P}(\mathbf{V}, b)$ exactly defines a polymatroid. The adaptive greedy algorithm reduces to the classical greedy algorithm of Edmonds (1970) that sorts jobs in order of non-increasing c_i. The optimality of the adaptive greedy algorithm leads naturally to a definition of *allocation indices* that characterise the optimal solutions of linear programs over extended polymatroids.

Definition 3 (Allocation Indices)

The allocation indices $\gamma_1, \ldots, \gamma_{|G \cup D|}$ *of the linear program (2.16) are given by*

$$\gamma_j = \sum_{k=j}^{|G \cup D|} \bar{y}^{\{\pi_1, \ldots, \pi_k\}}, \ j \in G \cup D,$$

where $(\bar{\mathbf{y}}, \pi)$ *is the output from the adaptive greedy algorithm.*

The allocation indices depend only upon the input $(\mathbf{V}, \mathbf{c})$ not the set function b. Bertsimas and Niño-Mora (1996) show that the dual solution $\bar{\mathbf{y}}$ is uniquely determined

by the input in that it is invariant in the way that ties are broken in the application of the algorithm, although the permutation π is not. The consistency in $\bar{\mathbf{y}}$ immediately translates to a similar consistency in the definition of the allocation indices.

Theorem 4 (Indexability)

(i) *The allocation indices in LP (2.16) are uniquely determined by* $\mathbf{c}$ *and* $\mathbf{V}$*;*

(ii) $\gamma_{\pi_1} \geq \gamma_{\pi_2} \geq \cdots \geq \gamma_{\pi_{|G \cup D|}}$.

Part (ii) follows from Definition 3 and states that for any permutation π that sorts the allocation indices in decreasing order, $\mathbf{v}(\pi)$ corresponds to an optimal solution of LP (2.16). Linear programs over extended polymatroids are said to have an *indexability* property. The next result follows easily from the above analysis and the duality theorem. We obey the convention that $\gamma_{|G \cup D|+1} = 0$.

Corollary 5 (Optimal Objective Value)

The optimal objective value Ψ *of the LP (2.16) is given by*

$$\begin{aligned} \Psi &= \sum_{j=1}^{|G \cup D|} \gamma_{\pi_j} \left(b(\pi_1, \ldots, \pi_j) - b(\{\pi_1, \ldots, \pi_{j-1}\}) \right) \\ &= \sum_{j=1}^{|G \cup D|} \left(\gamma_{\pi_j} - \gamma_{\pi_{j+1}} \right) b(\{\pi_1, \ldots, \pi_j\}). \end{aligned} \tag{2.17}$$

Bertsimas and Niño-Mora (1996) develop the achievable region approach by introducing a general concept of conservation laws. Their contribution for systems satisfying *generalised conservation laws* (GCL) extend the results on SCL in a natural way by providing a framework for tackling stochastic scheduling problems solved by priority index rules. If a performance vector satisfies GCL they are able to show that the achievable region is an extended polymatroid whose vertices correspond to strict priority policies.

A performance vector is said to satisfy GCL if a *weighted* sum of performances over all job classes $G \cup D$ is invariant under all admissible policies, and if, for all $S \subseteq G \cup D$ the

minimum *weighted* sum of performances over job classes in S is achieved by any policy that gives priority to job classes in S over job classes in S^c. More formally, we have the following.

Definition 4 (Generalised Conservation Laws)

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G \cup D|})$ *satisfies generalised conservation laws (GCL) if there exists a set function* $b : 2^{G \cup D} \to \mathbb{R}_+$ *and a matrix* $\mathbf{V} = (V_j^S)_{j \in G \cup D, S \subseteq G \cup D}$ *satisfying* $V_j^S > 0$ *for* $j \in S$, $S \subseteq G \cup D$, *such that, for all* $u \in \mathcal{U}$,

$$\sum_{j \in G \cup D} V_j^{G \cup D} x_j^u = b(G \cup D), \tag{2.18}$$

$$\sum_{j \in S} V_j^S x_j^u \geq b(S), \text{ for all } S \subset G \cup D, \tag{2.19}$$

and such that for all priority policies $\pi = \{\pi_1, \pi_2, \ldots, \pi_{|G \cup D|}\}$

$$\sum_{j=1}^{r} V_{\pi_j}^{\{\pi_1, \ldots, \pi_r\}} x_{\pi_j}^{\pi} = b(\{\pi_1, \ldots, \pi_r\}), \text{ for } r = 1, 2, \ldots, |G \cup D|. \tag{2.20}$$

We say the system satisfies $GCL(\mathbf{V}, b)$.

The following theorem, established by Bertsimas and Niño-Mora (1996), provides the connection between GCL and extended polymatroids.

Theorem 6 (Performance Region Characterisation)

If performance vector $\mathbf{x}$ *satisfies* $GCL(\mathbf{V}, b)$ *then:*

(i) *The performance vectors corresponding to strict priority policies are the vertices of* $\mathcal{P}(\mathbf{V}, b)$, *with* $\mathbf{x}^{\pi} = \mathbf{v}(\pi)$;

(ii) *The performance space* $\mathcal{X} = \mathcal{P}(\mathbf{V}, b)$.

If $V_i^S = 1$ for all $i \in S$, $S \subseteq G \cup D$ in Definition 4 then the system satisfies SCL. It is easy to see that parts (a) and (c) of Theorem 1 are a special case of Theorem 6.

We are now able to demonstrate the method of the achievable region approach applied to a general performance vector $\mathbf{x}$ that satisfies GCL. To achieve this we return to our original multi-class scheduling problem

$$\Psi = \inf_{u \in \mathcal{U}} \left(\sum_{j \in G \cup D} c_j x_j^u \right). \tag{2.21}$$

At the beginning of Section 2.3 we outlined the steps taken in applying the achievable region approach to a stochastic optimisation problem. Step (i) requires the identification of the performance space $\mathcal{X}$. From Theorem 6 we have that $\mathcal{X} = \mathcal{P}(\mathbf{V}, b)$. In step (ii) we reformulate the stochastic optimisation problem as a mathematical programming problem with feasible region $\mathcal{X}$. We solve the LP over $\mathcal{X}$ by running the adaptive greedy algorithm with input $(\mathbf{c}, \mathbf{V})$. Finally, in part (iii) we identify the scheduling policy that corresponds to the solution of the LP. We have that the solution to the LP lies at a vertex of the polyhedron $\mathcal{P}(\mathbf{V}, b)$ and takes the form $\mathbf{v}(\pi)$ which corresponds to the performance of the strict priority policy π.

The strong structural properties of extended polymatroids lead to strong structural properties in the scheduling problem. Suppose that we attach to job class j the index γ_j obtained from running the adaptive greedy algorithm with input $(\mathbf{c}, \mathbf{V})$. As a consequence of Theorem 4 an optimal policy will allocate priorities amongst the job classes in decreasing index order.

Theorem 7 (Indexability under GCL)

If a performance vector satisfies GCL then the stochastic scheduling problem given by (2.21) is solved by:

(i) *The priority policy π, i.e.;*

(ii) *The policy that selects at each decision epoch a job of currently largest index where π and γ are obtained from running the adaptive greedy algorithm with input* $(\mathbf{c}, \mathbf{V})$.

The following examples of Section 2.2 that describe models for local scheduling at each station in the network satisfy GCL:

Example 1 (The Klimov network)

To progress we introduce the following system parameters: The *total class j arrival rate* to the station, denoted Λ_j, is given by the solution to the system of linear equations

$$\Lambda_j = \lambda_j + \sum_{i \in G \cup D} \Lambda_i q_{ij}, \ j \in G \cup D.$$

For scheduling policies which give priority to job classes in $S \subseteq G \cup D$ the constants V_j^S are the *mean S-workload of a class j job* and are interpreted as the mean amount of processing required for a class j job to escape subset S for the first time, $j \in S$, $S \subseteq G \cup D$. These constants are computed by solving

$$V_j^S = \frac{1}{\mu_j} + \sum_{i \in S} q_{ij} V_i^S, \ j \in S, \ S \subseteq G \cup D.$$

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G \cup D|})$, with $x_j = E(N_j)/\mu_j$ satisfies GCL$(\mathbf{V}, b)$ with

$$b(S) = \frac{\sum_{j \in S} \frac{\Lambda_j}{\mu_j} V_j^S}{1 - \sum_{j \in S} \lambda_j V_j^S}, \ S \subseteq G \cup D.$$

Example 2 (The $M/M/1$ queue)

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G \cup D|})$, with $x_j = E(N_j)/\mu_j$ satisfies GCL$(\mathbf{V}, b)$ with

$$V_i^S = 1, \ i \in S, \ S \subseteq G \cup D \tag{2.22}$$

and

$$b(S) = \frac{\sum_{j\in S} \frac{\rho_j}{\mu_j}}{1 - \sum_{j\in S} \rho_j},\ S \subseteq G \cup D.$$

In (2.22) the constants $V_j^S = 1$ for all $j \in S,\ S \subseteq G \cup D$. Hence we can specify further that the performance vector satisfies SCL(b).

Example 3 (The $M/G/1$ queue)

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G\cup D|})$, with $x_j = E(N_j)/\mu_j$ satisfies GCL($\mathbf{V}, b$) with

$$V_i^S = 1,\ i \in S,\ S \subseteq G \cup D \tag{2.23}$$

and

$$b(S) = \left(\sum_{j\in G\cup D} \frac{\lambda_j M_j}{2}\right)\left(\frac{\sum_{j\in S}\rho_j}{1 - \sum_{j\in S}\rho_j}\right) + \sum_{j\in S}\frac{\rho_j}{\mu_j},\ S \subseteq G \cup D.$$

In (2.23) the constants $V_j^S = 1$ for all $j \in S,\ S \subseteq G \cup D$. Hence we can specify further that the performance vector satisfies SCL(b).

Example 4 (The $M/M/\eta$ queue)

The performance vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|G\cup D|})$, with $x_j = E(N_j)/\mu_j$ satisfies GCL($\mathbf{V}, b$) with

$$V_i^S = 1,\ i \in S,\ S \subseteq G \cup D \tag{2.24}$$

and

$$b(S) = \sum_{j\in S}\rho_j + \frac{\eta(\sum_{j\in S}\rho_j)^{\eta+1}}{\eta!(\eta - \sum_{j\in S}\rho_j)^2} \cdot \left[\sum_{r=0}^{\eta-1}\frac{(\sum_{j\in S}\rho_j)^r}{r!} + \frac{\eta(\sum_{j\in S}\rho_j)^\eta}{\eta!(\eta - \sum_{j\in S}\rho_j)}\right]^{-1},\ S \subseteq G \cup D.$$

In (2.24) the constants $V_j^S = 1$ for all $j \in S,\ S \subseteq G \cup D$. Hence we can specify further that the performance vector satisfies SCL(b).

2.6 Decomposability and Reducibility

We now highlight two important structural properties that are satisfied in a number of multi-class scheduling problems that satisfy GCL. Bertsimas and Niño-Mora (1996) introduced the notion of *decomposability*. Stochastic scheduling problems that exhibit this strong property, such as the examples we consider here, can be solved by breaking them down into a number of smaller sub-problems. *Reducibility*, introduced by Garbe and Glazebrook (1998), guarantees that systems formed by restricting service to different subsets of jobs are related to each other in a fairly simple way. Dacre and Glazebrook (2002) generalise the concept of reducibility, thus extending the results to cover systems outside the scope of previous analysis.

In what follows, we assume that the performance vector $\mathbf{x}$ satisfies GCL (2.18) - (2.20). We also suppose that we have a partition $\Delta = \{E_1, E_2, \ldots, E_K\}$ of the job set $G \cup D$, where

$$G \cup D = \bigcup_{E_k \in \Delta} E_k, \text{ and } E_k \cap E_l = \emptyset, \text{ for } k \neq l.$$

Definition 5 (Decomposability)

The system is decomposable with respect to the partition Δ if

$$V_j^S = V_j^{S \cap E_k}, \text{ for } j \in S \cap E_k,\ S \subseteq G \cup D.$$

The important consequence of decomposability is that the allocation indices of jobs in subset E_k depend *only* on the characteristics of jobs in that subset. If we define $\mathbf{V}^k = (V_j^S)_{j \in E_k, S \subseteq E_k}$ and $\mathbf{c}^k = (c_j)_{j \in E_k}$ then let $\boldsymbol{\gamma}^k$ be the index vector produced by the

adaptive greedy algorithm with input $(\mathbf{c}^k, \mathbf{V}^k)$.

Theorem 8 (Index Decomposition)

If the system is decomposable with respect to Δ *then*

$$\gamma_j = \gamma_j^k, \text{ for } j \in E_k, \ k = 1, \ldots, K.$$

For a proof of Theorem 8 see Bertsimas and Niño-Mora (1996).

Example 1

Klimov networks are decomposable with respect to the partition Δ provided that there is no routing between different partition sets. For this to be the case we must have

$$q_{ij} = 0 \text{ for } i \in E_k, \ j \notin E_k. \tag{2.25}$$

Examples 2-4

Systems satisfying SCL are trivially decomposable with respect to the partition Δ when $K = |G \cup D|$ and E_k is the singleton job set $\{k\}$ for $k = 1, 2, \ldots, |G \cup D|$. This follows from the fact that $V_j^S = 1$, for all $j \in S$ and $S \subseteq G \cup D$. Thus the $M/M/1$, $M/G/1$ and $M/M/\eta$ queueing systems considered here are all decomposable with respect to the partition Δ.

Shanthikumar and Yao (1992) show that the base function of SCL(b) systems is supermodular. This result does not hold in general for systems satisfying GCL. Garbe and Glazebrook (1998) utilise Theorem 8 to establish that decomposable GCL($\mathbf{V}, b$) systems exhibit a weaker form of supermodularity with respect to the partition Δ.

Definition 6 (Partial Supermodularity)

The set function $b : 2^{G \cup D} \to \mathbb{R}_+$ *is supermodular with respect to the partition* Δ *if, for*

all $\Sigma \subset \Delta$, *and subsets* S, T, σ *and* τ *satisfying*

$$S \subseteq T \subseteq \bigcup_{E_k \in \Sigma} E_k \text{ and } \sigma \subseteq \tau \subseteq \bigcup_{E_k \notin \Sigma} E_k,$$

we have

$$b(S \cup \tau) - b(S \cup \sigma) \leq b(T \cup \tau) - b(T \cup \sigma).$$

Lemma 9 (Partial Supermodularity in Decomposable Systems)
If a $GCL(\mathbf{V}, b)$ *system is decomposable with respect to* Δ, *then* $b : 2^{G \cup D} \rightarrow \mathbb{R}^+$ *is supermodular with respect to* Δ.

Garbe and Glazebrook (1998) define a reducibility property that is used to obtain results relating to the optimal station performance as a function of the set of job classes allowed access to service. This involves a reduction of the system from one that allows access to all job classes to a system that will only admit jobs that belong to subset of $G \cup D$ given by a union of partition sets, i.e.

$$S_\Sigma = \bigcup_{E_k \in \Sigma} E_k \text{ for some } \Sigma \subseteq \Delta.$$

Dacre and Glazebrook (2002) generalise this notion of reducibility extending the results to cover non-preemptive systems.

Definition 7 (Quasi-reducibility)
A system that satisfies $GCL(\mathbf{V}, b)$ *is quasi-reducible with respect to* Δ *if, for all* $\Sigma \subseteq \Delta$, *the reduced system that serves only jobs of class* $j \in \bigcup_{E_k \in \Sigma} E_k$ *satisfies* $GCL(\mathbf{V}_\Sigma, b_\Sigma)$, *where*

$$(V_\Sigma)_j^S = V_j^S,\ j \in S,\ S \subseteq S_\Sigma$$
$$b_\Sigma(S) = f(\Sigma) b(S),\ S \subseteq S_\Sigma.$$

where $f(\Sigma) : 2^{\Delta} \rightarrow \mathbb{R}_+$ *is a non-negative, non-decreasing, supermodular set function. We say that the system is fully reducible with respect to* Δ *in the special case of* $f(\cdot) \equiv 1$.

Example 1

The Klimov network is fully reducible with respect to the partition Δ when condition (2.25) is satisfied.

Examples 2 and 4

The $M/M/1$ and $M/M/\eta$ queueing systems are fully reducible with respect to Δ when $K = |G \cup D|$ and E_k is the singleton job set $\{k\}$, for $k = 1, 2, \ldots, |G \cup D|$.

Example 3

The $M/G/1$ queueing system is quasi-reducible with respect to Δ when $K = |G \cup D|$ and E_k is the singleton job set $\{k\}$, for $k = 1, 2, \ldots, |G \cup D|$, where the performance measure of a class j job is given by

$$x_j = \frac{E(N_j)}{\mu_j} - \frac{\lambda_j}{\mu_j^2}.$$

In this case

$$f(\Sigma) = \frac{\sum_{j \in S_\Sigma} \lambda_j M_j}{\sum_{j \in G \cup D} \lambda_j M_j}.$$

2.7 Properties of the Optimal Cost Function

Utilising decomposability and reducibility Dacre (1999) and Dacre and Glazebrook (2002) deduce structural properties of the station cost under an optimal scheduling regime, regarded as a function of the job classes allowed access to service. They are able to utilise these results to deduce properties of the optimal station cost regarded as a function of job arrival rates.

We now introduce structural results for reducible and decomposable systems for which we shall require the following notation and conventions. We denote the optimal set cost function by $\bar{\Psi}$. Without loss of generality, assume that jobs in $G \cup D$ are arranged in decreasing allocation index order, so $\gamma_1 \geq \ldots \geq \gamma_{|G \cup D|}$. The set $S^i = \{1, \ldots, i\}$ denotes the subset of $G \cup D$ comprising the i jobs with highest indices, and $S^0 = \emptyset$.

Before we present Theorem 11, that describes properties of the optimal set cost function, we introduce the following lemma concerning the calculation of $\bar{\Psi}(S_\Sigma)$. The reader is referred to Dacre (1999) for the proofs of these results.

Lemma 10

If the system is quasi-reducible and decomposable with respect to Δ, *then, for all* $\Sigma \subseteq \Delta$,

$$\bar{\Psi}(S_\Sigma) = f(\Sigma) \sum_{i=1}^{|G \cup D|} (\gamma_i - \gamma_{i+1}) b(S^i \cap S_\Sigma).$$

Theorem 11 (Supermodularity of Optimal Cost)

If the system is quasi-reducible and decomposable with respect to Δ, *then* $\bar{\Psi} : 2^\Delta \rightarrow \mathbb{R}_+$ *is non-decreasing, normalised and supermodular.*

Example 1

Dacre and Glazebrook (2002) introduce a novel re-labelling of job classes that allows the theory presented here to be applied to models with complex stochastic service structures such as the Klimov network. For the Klimov network we split job class j into $|G \cup D|$ sub-classes, $j = \{j_1, j_2, \ldots, j_{|G \cup D|}\}$. In this representation j_k denotes the set of current class j jobs that originally entered the system as class k jobs. Under this splitting of the job classes, jobs in sub-classes within j are physically indistinguishable from one another. We denote the set of jobs that originally entered the system as class j jobs by $E_j = \{1_j, \ldots, |G \cup D|_j\}$. Denote by E^2 the system comprising the set of job classes

$\{1_1, 1_2, \ldots, 1_{|G\cup D|}, 2_1, \ldots, |G \cup D|_{|G\cup D|}\}$. The external arrival rates given by

$$\lambda_{j_k} = \begin{cases} \lambda_j, & \text{if } j = k, \\ 0, & \text{otherwise;} \end{cases}$$

and routing probabilities

$$q_{j_k i_l} = \begin{cases} q_{ji}, & \text{if } k = l, \\ 0, & \text{otherwise.} \end{cases}$$

It is obvious that condition (2.25) is satisfied with respect to the partition $\bar{\Delta} = \{E_1, \ldots, E_{|G\cup D|}\}$. $\bar{\Delta}$ denotes the particular partition for this problem. Therefore, we have decomposability and (quasi-)reducibility with respect to $\bar{\Delta}$ and Theorem 11 applies.

The new system E^2 results from a simple relabelling of jobs from the original system. Removing the set E_j from the set of admissible jobs removes all external class j arrivals and the jobs they have evolved into. It follows that the optimal cost $\bar{\Psi}$ is a supermodular function of the set of jobs allowed *external* access to the system.

Examples 2-4

It follows from the discussion above that the $M/M/1$, $M/M/\eta$ and (after some manipulation) $M/G/1$ queueing systems are quasi-reducible and decomposable with respect to the partition Δ when $K = |G \cup D|$ and E_k is the singleton job set $\{k\}$, $k \in G \cup D$. We conclude that the optimal cost $\bar{\Psi}$ is a supermodular function of the set of jobs allowed access to the system.

When we have Poisson arrivals, job classes i and j, with identical cost and service characteristics and independent arrival processes, can be merged to form a single job class with arrival rate $\lambda_i + \lambda_j$. The converse is also true. Via standard Bernoulli splitting the arrivals of job class j, with arrival rate λ_j, can be classified to be one of type $j_1, \ldots, j_n$ with respective probabilities $p_{j_1}, \ldots, p_{j_n}$. The (nominally) distinct job classes $j_1, \ldots, j_n$

have identical cost and service characteristics and Poisson arrival rates $p_{j_1}\lambda_j, \ldots, p_{j_n}\lambda_j$ with $\sum_{r=1}^{n} p_{j_r} = 1$. This splitting of the arriving job classes is used by Dacre (1999) and Dacre and Glazebrook (2002) to translate between properties of the optimal set cost function $\bar{\Psi}$ and the optimal cost function Ψ, a function of the job arrival rate vector.

The following lemmas characterise the optimal return $\Psi(\boldsymbol{\lambda}^G)$, as we should expect, we find that Ψ is increasing in job arrival rates.

Lemma 12 (Supermodularity and Monotonicity of Ψ)

For any feasible arrival rate vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}_+^{|G \cup D|}$

$$\Psi(\mathbf{x}+\mathbf{y}+\mathbf{z}) - \Psi(\mathbf{x}+\mathbf{y}) \geq \Psi(\mathbf{x}+\mathbf{z}) - \Psi(\mathbf{x}), \tag{2.26}$$

$$\Psi(\mathbf{x}+\mathbf{y}) \geq \Psi(\mathbf{x}).$$

Lemma 13 (Continuity of Ψ)

Ψ is a continuous function on the set of feasible arrival rates.

We are now in a position to consider the effect of the preceding discussion on the full routing problem and we restore the station suffix m. The static routing problem of interest is given by the optimisation problem (2.3). Under this representation the routing problem is in fact a joint routing/local scheduling problem. Key to the solution of (2.3) is the characterisation of the optimal station costs Ψ_m as functions of the generic load $\boldsymbol{\lambda}_m^G$. For such local scheduling problems the achievable region approach has had notable success. The station models we have focused our analysis upon, namely the Klimov network, the $M/M/1$ queue, the $M/G/1$ queue and the $M/M/\eta$ queue, all satisfy the generalised conservation laws of Bertsimas and Niño-Mora (1996) in Definition 4. The optimal scheduling policy $u_m \in \mathcal{U}_m$ for systems satisfying GCL is a strict priority index policy whose indices may be computed from a single run of the adaptive greedy algorithm. Further, the station models exhibit the important properties of decomposability and (quasi-)reducibility. For decomposable and (quasi-)reducible systems it is possible

to deduce properties of the optimal station cost as a function of the job classes allowed access to service, $\bar{\Psi}_m$. The characteristics of the $\bar{\Psi}_m$ may in turn be used to deduce properties of the optimal station cost as a function of generic job arrival rates, $\Psi_m(\boldsymbol{\lambda}_m^G)$, a prime objective in the analysis of the static routing problem. For the purposes of solving the routing problem in (2.3) we would ideally like each $\Psi_m(\boldsymbol{\lambda}_m^G)$ to be an increasing and convex function of the offered generic load $\boldsymbol{\lambda}_m^G$. Later we shall show that we have full convexity in a number of special cases, however, in higher dimensions, full convexity is a very strong property and in general we must settle for the weaker form described in Definition 8 in which convexity is available in certain directions (NE-SW) only in load space.

Definition 8 (North-East Convexity)

A function $f : D \to \mathbb{R}$, where $D \subseteq \mathbb{R}^n$ is convex, is North-East (NE) convex *if, for all $\alpha \in [0,1]$ and all $\mathbf{x}, \mathbf{y} \in D$ such that $\mathbf{x} - \mathbf{y} \in \mathbb{R}^n_+ \cup \mathbb{R}^n_-$,*

$$\alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) \geq f(\alpha \mathbf{x} + (1-\alpha)\mathbf{y}).$$

Theorem 14 (North-East Convexity of Ψ_m)

Ψ_m is North-East convex on the set of feasible arrival rates.

Proof

For any feasible arrival rate vectors $\mathbf{x}, \mathbf{y}$ satisfying $\mathbf{x} - \mathbf{y} \in \mathbb{R}_+^{|G \cup D|}$, and any $p \in \mathbb{N} \cup \{0\}$, $q \in \mathbb{N}$ such that $p \leq q$, we have

$$\begin{aligned} &\Psi_m(\mathbf{x}) - \Psi_m\left(\mathbf{x} + \frac{p}{q}(\mathbf{y} - \mathbf{x})\right) \\ &\qquad = \sum_{r=1}^{p} \left[\Psi_m\left(\mathbf{x} + \frac{r-1}{q}(\mathbf{y} - \mathbf{x})\right) - \Psi_m\left(\mathbf{x} + \frac{r}{q}(\mathbf{y} - \mathbf{x})\right)\right] \end{aligned}$$

$$\geq p\left[\Psi_m\left(\mathbf{x}+\frac{p-1}{q}(\mathbf{y}-\mathbf{x})\right)-\Psi_m\left(\mathbf{x}+\frac{p}{q}(\mathbf{y}-\mathbf{x})\right)\right] \tag{2.27}$$

$$\geq \frac{p}{q-p}\sum_{r=p+1}^{q}\left[\Psi_m\left(\mathbf{x}+\frac{r-1}{q}(\mathbf{y}-\mathbf{x})\right)-\Psi_m\left(\mathbf{x}+\frac{r}{q}(\mathbf{y}-\mathbf{x})\right)\right] \tag{2.28}$$

$$=\frac{p}{q-p}\left[\Psi_m\left(\mathbf{x}+\frac{p}{q}(\mathbf{y}-\mathbf{x})\right)-\Psi_m(\mathbf{y})\right],$$

where inequalities (2.27) and (2.28) follow from property (2.26) of Lemma 12. Rearranging this expression gives

$$\left(1-\frac{p}{q}\right)\Psi_m(\mathbf{x})+\frac{p}{q}\Psi_m(\mathbf{y})\geq\Psi_m\left(\left(1-\frac{p}{q}\right)\mathbf{x}+\frac{p}{q}\mathbf{y}\right),$$

and because our choices of p and q were arbitrary, we have

$$(1-\alpha)\,\Psi_m(\mathbf{x})+\alpha\Psi_m(\mathbf{y})\geq\Psi_m\left((1-\alpha)\,\mathbf{x}+\alpha\mathbf{y}\right) \tag{2.29}$$

for all $\alpha\in\mathbb{Q}\cap[0,1]$.

As $\mathbb{Q}\cap[0,1]$ is a dense subset of $[0,1]$, then by the continuity of Ψ_m we have that (2.29) holds for all $\alpha\in[0,1]$ and the result follows. □

On the basis of Lemma 13 and Theorem 14 we may assert the existence of a solution to the routing problem in (2.3). We use $\tilde{\Phi}:\mathbb{R}^{|G|\times|\mathcal{M}|}\rightarrow\mathbb{R}$ to denote the system cost as a function of generic machine arrival rates $\Lambda^G=\{\lambda_m^G\}_{m\in\mathcal{M}}$, i.e.,

$$\tilde{\Phi}_{\mathcal{M}}(\Lambda^G)=\sum_{m\in\mathcal{M}}\Psi_m(\lambda_m^G).$$

We also use $F\subseteq\mathbb{R}^{|G|\times|\mathcal{M}|}$ to denote the feasible solution space for (2.3).

Corollary 15 (Existence of Solutions to the Static Routing Problem)

The infimum in the routing problem (2.3) is attained.

Proof

The continuity of $\tilde{\Phi}_{\mathcal{M}}$ on F follows immediately from the continuity of the Ψ_m, $m \in \mathcal{M}$, asserted in Lemma 13. Let c be the cost of some feasible solution, Λ^G, to the routing problem. Consider the set $F' = \tilde{\Phi}_{\mathcal{M}}^{-1}([0, c])$. Plainly, $\Lambda^G \in F' \subseteq F$ and the set F is bounded by stability constraints. Hence F' is non-empty and bounded. From the continuity of $\tilde{\Phi}_{\mathcal{M}}$ and the fact that $[0, c]$ is closed in $\mathbb{R}$ we then have that F' is closed in $\mathbb{R}^{|G|\times|\mathcal{M}|}$. A continuous function on a closed and bounded subset of $\mathbb{R}^{|G|\times|\mathcal{M}|}$ attains its minimum and the result follows. □

2.8 Special Cases of the Static Routing Problem

We now highlight two special scenarios, introduced by Dacre (1999), that allow us to say a great deal about the solutions to our routing problem. The first scenario involves cases in which the optimal cost function at each station is fully convex in generic arrival rates. This in turn leads to the full convexity of the optimal system cost function. The second scenario of interest is one in which the network comprises homogeneous stations. In such situations a simple, intuitive routing policy has been shown to be optimal in a fully convex system and is never 'orders of magnitude' worse than optimal more generally. In what follows we assume that all dedicated arrival rates are fixed and known.

2.8.1 Static Routing in Fully Convex Systems

We shall now describe three special cases in which the directional convexity of Ψ_m, the optimal station cost, described in Theorem 14 may be strengthened to full convexity. The first case yielding full convexity in the station returns occurs in situations in which there is a single class of generic job. Taking $n = 1$ in Definition 8 yields (full) convexity for the function f. The following result is an immediate consequence of the structural properties of Ψ_m given by Theorem 14.

Corollary 16 (Full Convexity: Case 1)

When $|G| = 1$, Ψ_m *is a continuous, non-decreasing and convex function of the generic load* λ_m^G.

The second general case arises in situations in which the nature of the stochastic evolution of the process at a station is independent of the class membership of the (generic) jobs chosen for processing. We apply the term *stochastically indistinguishable* to generic jobs in such situations. In Examples 1-4 we require the following at each station m:

Example 1

When each station is modelled as Klimov network we require $\mu_{jm} = \mu_m$, $j \in G$, and $q_{ik}^m = q_{jk}^m = q_k^m$ for all choices of i, j and $k \in G$;

Example 2

When each station is modelled as a $M/M/1$ system we require $\mu_{jm} = \mu_m$, $j \in G$;

Example 3

When each station is modelled as a $M/G/1$ system we require $\mu_{jm} = \mu_m$, $j \in G$ and $M_{jm} = M_m$, $j \in G$ and $D_m = \emptyset$;

Example 4

When each station is modelled as a $M/M/\eta$ system service requirements are i.i.d. and hence job classes are already indistinguishable.

While the notion of stochastic indistinguishability applies to each of the station models of interest we must note that Corollary 17 holds for Examples 1, 2 and 4 only. For a proof of this result see Dacre (1999).

Corollary 17 (Full Convexity: Case 2 - Examples 1, 2 and 4 only)

If generic job classes are stochastically indistinguishable at each station then Ψ_m is a continuous, non-decreasing and convex function of the generic load λ_m^G.

The final case which yields full convexity in the optimal return at station m is one in which the system controller is unable to distinguish between jobs from different classes. Here the controller routing probabilities are constrained to be class-independent - i.e., $p_{jm} = p_m,\ j \in G, m \in \mathcal{M}$. The routing problem in (2.3) now becomes

$$\min \sum_{m \in \mathcal{M}} \Psi_m \left(p_m \lambda^G\right) \tag{2.30}$$
$$\text{such that } \sum_{m \in \mathcal{M}} p_m = 1,\ p_m \geq 0,\ m \in \mathcal{M}.$$

Following (2.30), our interest is now in the functions $\Psi_m^* : [0,1] \rightarrow \mathbb{R}_+$ given by $\Psi_m^*(p) = \Psi_m(p\lambda^G)$ with λ^G fixed. The following is an immediate consequence of Theorem 14. See also Definition 8.

Corollary 18 (Full Convexity: Case 3)

The function Ψ_m^* is continuous, non-decreasing and convex.

We now point to a major implication of the convexity results for Cases 1-3 above.

Theorem 19 (Convexity of the Optimal Return for the System)

When Ψ_m is convex for all $m \in \mathcal{M}$ then $\Phi_{\mathcal{M}}$, the optimal return for the system, is convex in the system generic arrival rate λ^G.

Proof

Consider feasible generic arrival rates $\lambda^G,\ \lambda'^G \in \mathbb{R}_+^{|G|}$ with associated optimal solutions

to (2.3) given by $(\lambda_m^{*G})_{m\in\mathcal{M}}$ and $(\lambda_m^{*'G})_{m\in\mathcal{M}}$ respectively. Hence for any $\alpha \in [0,1]$,

$$\alpha\Phi_{\mathcal{M}}(\lambda^G) + (1-\alpha)\Phi_{\mathcal{M}}(\lambda'^G) = \sum_{m\in\mathcal{M}} \left\{\alpha\Psi_m(\lambda_m^{*G}) + (1-\alpha)\Psi_m(\lambda_m^{*'G})\right\}$$

$$\geq \sum_{m\in\mathcal{M}} \Psi_m\left\{\alpha\lambda_m^{*G} + (1-\alpha)\lambda_m^{*'G}\right\} \tag{2.31}$$

$$\geq \Phi_{\mathcal{M}}\left\{\alpha\lambda^G + (1-\alpha)\lambda'^G\right\}, \tag{2.32}$$

as required. Inequality (2.31) utilises the convexity of Ψ_m, $m \in \mathcal{M}$, and inequality (2.32) the feasibility of $\left\{\alpha\lambda_m^{*G} + (1-\alpha)\lambda_m^{*'G}\right\}_{m\in\mathcal{M}}$ as a solution to the problem with generic rate $\alpha\lambda^G + (1-\alpha)\lambda'^G$. □

The three special cases described above in which Ψ_m, the optimal station cost, is fully convex considerably ease the task of solving the full static routing problem in (2.3). Subject to smoothness conditions (which in our Examples will be met), a number of standard numerical routines will yield optimal solutions. Simple algorithms are available in the case where $|G| = 1$ and the optimal cost at each station is increasing, convex and differentiable in the arrival rate. Dacre (1999) describes a gradient matching algorithm, derived from Tantawi and Towsley (1985), which yields solutions to the static routing problem when the Ψ_m are differentiable. Furthermore, Dacre (1999) has shown that solutions to the $|G| = 1$ case exhibit a monotonicity property which, in the absence of dedicated jobs, allows for the solution of a routing problem in which there are $|G|$ stochastically indistinguishable generic job classes. Such problems can be solved by considering $|G|$ related problems of routing a single generic job class over the system. An index policy determining the optimal schedule reduces to one which assigns priority to the job class with the highest holding cost rate.

2.8.2 Static Routing when Stations are Homogeneous

The results presented so far make no assumptions about the *relative* processing capabilities of different stations. We now consider the important special case of *homogeneous stations*

in which the optimal return function Ψ_m is the same for all $m \in \mathcal{M}$. In this case the Equal Splitting Policy (ESP), $p_{jm} = 1/M,\ j \in G,\ m \in \mathcal{M}$, in which the load is divided equally between the stations, is optimal when the returns are convex.

Theorem 20 (Optimal Policy: Homogeneous Stations with Convex Returns)
When stations are homogeneous and the optimal return Ψ at each station is continuous, Ψ is also convex if and only if ESP is optimal for the routing problem for all generic arrival rates $\boldsymbol{\lambda}^G$.

Proof

Suppose that Ψ is convex. For any generic load $\boldsymbol{\lambda}_m^G$ assigned to station m we have

$$\sum_{m \in \mathcal{M}} \Psi\left(\boldsymbol{\lambda}_m^G\right) \geq M\Psi\left(\sum_{m \in \mathcal{M}} \frac{1}{M}\boldsymbol{\lambda}_m^G\right) = M\Psi\left(\frac{1}{M}\boldsymbol{\lambda}^G\right). \tag{2.33}$$

The final term is plainly the cost associated with ESP which is therefore optimal for all generic loads. Conversely, if ESP is optimal for all loads then we have the first inequality in (2.33) holding for all feasible $\boldsymbol{\lambda}_m^G,\ m \in \mathcal{M}$. Coupled with the continuity of Ψ, this is sufficient for its convexity. □

ESP seems a natural routing policy when stations are homogeneous. However, when we leave the convex case of Theorem 20 ESP is not guaranteed to be optimal. There are situations in which the optimal return Ψ_m can be concave in directions other than NE-SW. In such cases routing policies where stations specialise in processing a subset of job classes may be favoured.

In the general case for homogeneous stations in which Ψ is assumed to be NE-convex only, Theorem 20 is replaced by the much weaker Theorem 21.

Theorem 21 (Optimal Routing when Stations are Homogeneous)
When the optimal return Ψ at each station is North-East convex, there is an optimal routing policy, $\mathbf{P}^*$, in which there are no stations $i,\ j$ for which $\boldsymbol{\lambda}_i^G - \boldsymbol{\lambda}_j^G \in \mathbb{R}_+^{|G|}$, i.e. no

station is assigned more of all generic job classes than any other station.

Proof

If $\boldsymbol{\lambda}_1^G$, the generic arrival rate assignment to station 1, and $\boldsymbol{\lambda}_2^G$, the assignment to station 2 are such that $\boldsymbol{\lambda}_1^G - \boldsymbol{\lambda}_2^G \in \mathbb{R}_+^{|G|}$ then, by Definition 8 and the North-East convexity of Ψ we have

$$\Psi\left(\boldsymbol{\lambda}_1^G\right) + \Psi\left(\boldsymbol{\lambda}_2^G\right) \geq 2\Psi\left(\frac{1}{2}\boldsymbol{\lambda}_1^G + \frac{1}{2}\boldsymbol{\lambda}_2^G\right).$$

Hence the system cost cannot be increased by splitting the load evenly between the two stations. □

If we regard the determination of the optimal routing policy as a matter of minimising a function of $|G| \times (M-1)$ variables, namely

$$\tilde{\Phi}_{\mathcal{M}}\left(\boldsymbol{\lambda}_1^G, \ldots, \boldsymbol{\lambda}_{M-1}^G\right) = \sum_{m=1}^{M-1} \Psi(\boldsymbol{\lambda}_m^G) + \Psi(\boldsymbol{\lambda}^G - \sum_{m=1}^{M-1} \boldsymbol{\lambda}_m^G) \tag{2.34}$$

then the next result follows from a routine application of the standard Lagrangian method.

Lemma 22 (ESP a Stationary Point)

When Ψ is differentiable, the member of $\mathbb{R}_+^{|G|\times(M-1)}$ corresponding to ESP, namely $\boldsymbol{\lambda}_m^G = \frac{1}{M}\boldsymbol{\lambda}^G$, $1 \leq m \leq M-1$, is a stationary point of $\tilde{\Phi}_{\mathcal{M}}$.

The North-East convexity of Ψ rules out the stationary point in Lemma 22 being a (local) maximum. However ESP can occur at a (local) minimum or a saddle point. Results from an extensive numerical investigation into the performance of ESP by Dacre (1999) suggest that for problems close to the conditions which yield Theorem 21 ESP performs well. More generally, ESP can be significantly sub-optimal for some problem instances, however, it is never 'orders of magnitude' worse than optimal. Encouraged by the performance of ESP Dacre (1999) was able to establish suboptimality bounds for

two important cases, namely, the multi-class $M/M/1$ queueing model with preemptive scheduling controls of Example 1 and a special case of Example 3, the multi-class $M/G/1$ queueing system with non-preemptive scheduling controls, in which service requirements are exponentially distributed.

In conducting his computational investigation, Dacre (1999) applies an exhaustive search technique to obtain optimal solutions for each problem within his study. When evaluating the optimal routing policy for larger problems, even of reasonable size, such methods may not be computationally feasible. For problems of the form (2.34) there exist heuristic optimisation methods that deal with the problems created by local minima with varying degrees of success. One approach is to apply standard local optimisation techniques using a range of starting points. More recently developed heuristic techniques have been proved to successfully solve similar large scale optimisation problems. See, for example, the overview of simulated annealing by Brooks and Morgan (1985). However, in convex cases any local minimum will be a global minimum and there are a number of search techniques that are guaranteed to quickly converge to the optimal solution. For many of the problems investigated by Dacre (1999) the objective came close to being convex. He describes a Linear Programming Heuristic that sequentially solves a series of Linear Programs over a feasible region that are further restricted at each step by the addition of a new constraint which (in the case of full convexity) is a supporting hyperplane of the objective. The result of this is a sequence of increasingly accurate piecewise linear approximations to the objective near its minimum. The algorithm terminates when a suitable stopping criterion is satisfied. The heuristic is shown to have strong performance and improves upon ESP. However one must note that the quality of the improvement deteriorates as Ψ_m becomes substantially non-convex. That said, the heuristic's performance suggests that it is a reasonable approach to routing problems in the general case of heterogeneous stations for cases in which the optimal returns are close to convex.

2.9 Routing Jobs to Specialised Stations

The material covered in the preceding sections provides an excellent framework which can be applied to the analysis of queueing control problems of the general structure outlined in Section 2.2. Here we shall investigate a routing problem proposed by Becker et al. (2000) who consider a network of heterogeneous stations. Each station possesses a degree of specialisation in relation to certain a job class, in that the station is (uniquely) equipped to provide the most efficient service to jobs from that job class amongst all stations within the network. Becker at al.'s work was motivated primarily by problems in call centres seeking to provide technical support or service for a range of products. In the call centre, customer enquiries about a particular product are routed to operators who will process the call. Ideally, an enquiry will be routed to an operator who has expertise in providing service on the product in question. However, it may be that the company is receiving a high volume of calls concerning new products. In such circumstances routing all calls to the expert operators may induce long waiting times for enquiries on these products. It may be beneficial, then, to route some of these requests to operators with less expertise who may, for example, take longer to field the enquiry.

Keeping with the theme of the current chapter we retain our focus of considering only static routing policies. In the context of the current routing problem such static policies are a viable option in situations where the required state information (at each decision epoch) for the implementation of a dynamic routing policy is unavailable or is simply too expensive to operate. We must note that if this state information were available one would expect dynamic policies to considerably improve system performance. However, the development of such dynamic policies is a challenging and involved problem and is the focus of the following chapters. Becker et al. (2000) adopted the same approach to static routing as developed here. Static policies use simple Bernoulli routing probabilities for the assignment of arriving jobs to the stations within the network, which allows each station to be analysed as a set of independent queues. Their study compared the performance of static routing policies when the stations schedule their work optimally with alternative

scheduling strategies commonly utilised in the call centre environment.

In what follows we present the model considered by Becker et al. (2000) which is a particular example of the general model introduced in Section 2.2. We conduct a numerical study in which we shall reproduce the results of Becker et al. (2000) and then extend their study to cover a set of problems which consider a range of system set ups. The set up used by Becker et al. (2000) and in our extension has a specialised station available for processing each job class. We further extend our consideration of this problem to envisage situations in which the system controller is restricted by a limited budget. Under such a restriction the controller may experience a reduction in the total number of stations available with which to process arriving calls. However, for any given set up we allow the controller to employ any possible configuration of specialist stations (including multiple specialist station types). We conduct a numerical study to assess what mix of expertise will provide optimum system performance given the budget limitations.

2.9.1 The Problem

We present the call centre problem as an instance of the general static routing problem presented in Section 2.2. The system comprises a set $\mathcal{M} = \{1, 2, \ldots, M\}$ of stations each modelled as the multi-class $M/G/1$ queueing system of Example 3, Section 2.2. Generic jobs arrive at the system controller as a Poisson process with (total) rate λ. Each arriving generic job is of class g with independent probability a_g, $g \in G$ where $\sum_{g \in G} a_g = 1$. We assume that there is no dedicated traffic. Upon arrival to the system controller generic job g is assigned to station m with fixed probability p_{gm}, $m \in \mathcal{M}$. To be a valid set of routing probabilities the controller routing matrix $\mathbf{P} = (p_{gm})_{g \in G, m \in \mathcal{M}}$ must satisfy $\mathbf{Pe} = \mathbf{e}$ where $\mathbf{e}$ is a $|G|$-vector of 1's so that each arriving job is routed to a station with probability 1. Let $\lambda_{gm} = \lambda a_g p_{gm}$ denote the class g arrival rate to station m. Queueing jobs at station m have general service requirements which are i.i.d. within each class. Class g service times have mean μ_{gm}^{-1} and finite second moment M_{gm}, $g \in G$.

Each station m operates an admissible scheduling policy $u_m \in \mathcal{U}_m$. Many call centres employ a *first come first served* (FCFS) scheduling policy in the belief that processing calls in their arrival order is ultimately a "fair" way to treat their customers. However, significant improvements to overall performance can be made if the incoming calls to a particular station are scheduled optimally. We shall examine the system performance under both scheduling strategies. For notational convenience we shall use the single letter abbreviations F and O to denote the FCFS and the optimal scheduling policy respectively.

In the context of the current problem the aim of any routing/scheduling policy will be to reduce the call lengths (waiting and service time) experienced by the call centre customers. A natural performance measure in these circumstances will be the total expected delay. Each job class $g \in G$ is associated with a nonnegative *delay weight*, c_g, which can be thought of as some form of penalty incurred for the delay of jobs in class g. The total expected long-run *weighted* delay (per unit time) is given by

$$\sum_{m \in \mathcal{M}} \sum_{g \in G} \lambda_{gm} c_g E_{u_m}(D_{gm}),$$

where D_{gm} is the delay experienced by a class g job at station m and the expectation is taken in steady state with respect to scheduling policy u_m. Our goal is to determine the 'best' controller routing matrix $\mathbf{P}_u$ under admissible scheduling policy u for fixed λ minimising the total expected long-run weighted delay, denoted by $\widehat{\Phi}^u(\mathbf{P}_u)$. This problem is given by the minimisation

$$\min_{\mathbf{P}_u} \widehat{\Phi}^u(\mathbf{P}_u). \tag{2.35}$$

We denote the controller routing matrix achieving the minimum in (2.35) by $\mathbf{P}_u^*$. Note that in order for the system to conform with an appropriate notion of stability we require

the traffic intensity at each station m, given by

$$\rho_m = \sum_{g=1}^{|G|} \rho_{gm} = \sum_{g=1}^{|G|} \lambda_{gm}/\mu_{gm},\ m \in \mathcal{M}, \tag{2.36}$$

to be less than 1. In (2.36) we use ρ_{gm} to denote the traffic intensity at station m due to class g jobs.

We now proceed to develop expressions for the total expected long-run weighted delay under the two scheduling policies F and O.

Under policy O for fixed λ and from (2.3) the routing problem given by (2.35) can be expressed as

$$\widehat{\Phi}^O(\mathbf{P}_O) = \sum_{m=1}^{M} \widehat{\Psi}_m^O(\mathbf{P}_O),$$

where $\widehat{\Psi}_m^O(\mathbf{P}_O)$ is the total expected weighted delay at station m under policy O. From Little's result it is well known that $E(N_{gm}) = \lambda_{gm}E(D_{gm})$. Hence, by following the approach to local scheduling outlined in Sections 2.3-2.5 we have that for the $M/G/1$ queueing system station model policy O is a strict priority policy which is applied non-preemptively. Further, policy O is the so-called $c\mu$-rule which takes the following simple form for this problem: renumber the job classes such that

$$c_1\mu_{1m} \geq c_2\mu_{2m} \geq \ldots \geq c_{|G|}\mu_{|G|m}.$$

The optimal policy is for station m to serve the job with the currently smallest job class identifier at each service completion epoch for processing. From Corollary 5 the total expected long-run weighted delay at station m under policy O is given by

$$\widehat{\Psi}_m^O(\mathbf{P}_O) = \sum_{g=1}^{|G|} (c_g\mu_{gm} - c_{g+1}\mu_{g+1m}) b_m(\{1,\ldots,g\}),\ m \in \mathcal{M}, \tag{2.37}$$

where

$$b_m(S) = \left(\sum_{g\in G} \frac{\lambda_{gm} M_{gm}}{2}\right)\left(\frac{\sum_{g\in S}\rho_{gm}}{1-\sum_{g\in S}\rho_{gm}}\right) + \sum_{g\in S}\frac{\rho_{gm}}{\mu_{gm}},\ S\subseteq G.$$

Under scheduling policy F the well known Pollaczek-Khintchine-Kendall formula provides the expected long-run waiting time $E(W)$ for a nonpreemptive $M/G/1$ queueing system. Simple calculations yield that, for the single station problem, the expected long-run waiting time at station m (i.e. expected waiting time in the queue only) is

$$E_F(W_m) = \frac{1}{2}\sum_{g\in G}\frac{\lambda_{gm}M_{gm}}{1-\rho_m},\ m\in\mathcal{M}.$$

The total expected delay (i.e. expected waiting time in the queue and expected service time) for job class g is then given by

$$E_F(D_g) = \sum_{m\in\mathcal{M}} p_{gm}(E_F(W_m) + \mu_{gm}^{-1}),\ g\in G.$$

The total expected long-run weighted delay under policy F is

$$\begin{aligned}\widehat{\Phi}^F(\mathbf{P}_F) &= \sum_{g\in G}\lambda a_g c_g E_F(D_g)\\ &= \sum_{g\in G}\lambda a_g c_g\left[\sum_{m\in\mathcal{M}} p_{gm}(E_F(W_m)+\mu_{gm}^{-1})\right].\end{aligned} \tag{2.38}$$

For any $\mathbf{P}$ resulting in a stable system under both scheduling policies it is clear that $\widehat{\Phi}^O(\mathbf{P}) \leq \widehat{\Phi}^F(\mathbf{P})$. However, policy O requires the knowledge of the composition of the queueing jobs at each station and as such would be logistically more complex to apply in the call centre setting than policy F. It is then unclear as to the benefit gained by switching to this optimal scheduling strategy O. We now present the results to a computational study comparing system performance when the stations follow the two approaches mentioned here.

2.9.2 Numerical Study

Our numerical investigation shall initially consider a collection of problems in which we have an equal number of stations and jobs classes. These problem collections range from 4 stations with 4 job classes to 9 stations with 9 jobs classes. In the course of this study we reproduce the results from the computational study of Becker et al. (2000) in which they consider a system of 5 stations with 5 job classes and then relate their findings to the range of problems considered here.

For every problem considered we assume that class g jobs at station m have a gamma service time distribution with scale parameter α_m and shape parameter $k_{gm}\beta_m$. The first two moments of the service time distribution are given by

$$\mu_{gm}^{-1} = \frac{k_{gm}\beta_m}{\alpha_m}$$

and

$$M_{gm} = \frac{\{(k_{gm}\beta_m)^2 + k_{gm}\beta_m\}}{\alpha_m^2},$$

where we take

$$k_{gm} = |g - m| + 1.$$

Each station has the capability to process jobs from each job class $g \in G$. We say that station m is a specialist processing station for job class $g = m$, $g \in G$. It is clear that in this case the first two moments of service time for g-jobs are at their smallest, hence, such g-jobs are processed most expediently at this station.

The systems and their defining parameters are as follows:

System 1. $M = 4$, $|G| = 4$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (4^{-1} + 0.25) - 0.1g$ for $g = 1, \ldots, 4$.

System 2. $M = 5$, $|G| = 5$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (5^{-1} + 0.15) - 0.05g$ for $g = 1, \ldots, 5$.

System 3. $M = 6$, $|G| = 6$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (6^{-1} + 0.175) - 0.05g$ for $g = 1, \ldots, 6$.

System 4. $M = 7$, $|G| = 7$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (7^{-1} + 0.12) - 0.03g$ for $g = 1, \ldots, 7$.

System 5. $M = 8$, $|G| = 8$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (8^{-1} + 0.135) - 0.03g$ for $g = 1, \ldots, 8$.

System 6. $M = 9$, $|G| = 9$, $\alpha_m = 2$, $\beta_m = 1$ and $c_g = 1$. The g-job system arrival probabilities are given by $a_g = (9^{-1} + 0.1) - 0.02g$ for $g = 1, \ldots, 9$.

System 2 is exactly the system considered by Becker et al. (2000). In extending their study to cover a wider range of system set ups we have retained their original parameter values for simple comparison. Note that given the a_g's for each system have been arbitrarily chosen so that job class 1 arrives at the system most frequently and job $|G|$ the least frequently.

To progress we must determine the maximal *feasible* total system arrival rate, $\bar{\lambda}$, ensuring a stable system (i.e. $\rho_m < 1$, $m \in \mathcal{M}$) for Systems 1-6 above. This is achieved by solving the following maximisation problem:

$$\bar{\lambda} = \max_{\lambda,\, \mathbf{P}} \lambda \tag{2.39}$$

such that

$$\lambda \sum_{g=1}^{|G|} p_{gm} a_g \mu_{gm}^{-1} \leq 1,\ m \in \mathcal{M},$$

$$\sum_{m=1}^{M} p_{gm} = 1,\ g \in G,$$

$$p_{gm} \geq 0,\ g \in G,\ m \in \mathcal{M}.$$

Given $\bar{\lambda}$ we can investigate the performance of the scheduling policies for a range of $\lambda < \bar{\lambda}$ representing low to high traffic loads.

For a given feasible λ we can determine the optimal controller routing matrix $\mathbf{P}_u^*$ under scheduling policies F and O by solving the optimisation problem in (2.35). In order to solve (2.35) we formulate the following optimisation problem:

$$\min_{\mathbf{P}_u} \widehat{\Phi}^u(\mathbf{P}_u) \tag{2.40}$$

such that

$$\lambda \sum_{g=1}^{|G|} p_{gm} a_g \mu_{gm}^{-1} \leq 1 - \epsilon, \ m \in \mathcal{M}, \tag{2.41}$$

$$\sum_{m=1}^{M} p_{gm} = 1, \ g \in G,$$

$$p_{gm} \geq 0, \ g \in G, \ m \in \mathcal{M}.$$

The objective in (2.40) for policy O is given by (2.37) and by (2.38) for policy F. Note that the given representation of stability constraint (2.41) is required for the problem to be numerically implementable. In every problem studied we set $\epsilon = 0.01$. To solve the optimisation problem in (2.40) for both scheduling policies we utilise NAG routine E04UCF, an optimisation routine that uses a sequential quadratic programming method to achieve the minimum in (2.40). Given the fact that the optimal returns, $\widehat{\Phi}^u(\mathbf{P}_u)$, under both policies are typically non-convex the optimisation routine E04UCF can only guarantee locally optimal solutions. Following the comments at the end of Section 2.8.2, our approach to obtaining a good solution to the optimisation problem is to run the optimisation routine from 10 randomly generated starting points, with each run possibly generating a new locally optimal solution to the routing problem. We take as our solution to the optimisation problem the minimum objective value from the 10 program runs.

We now report the findings of the original study by Becker et al. (2000) which investigated the performance of System 2. For our extended study we shall appropriately

generalise these findings in relation to Systems 1-6 above.

(i) Tables 2.1, 2.2 contain the optimal routing probabilities under policies F and O, respectively, for a specific problem for System 2 in our larger study (see Tables 2.4 and 2.10), in which the total system arrival rate λ is set at 7.88 (equivalently 95% of $\bar{\lambda} = 8.29$). The total expected delay under F is 89.47 and is 66.67 under O. Station 1 is the specialist processing station for job class 1 but is unable to process all class 1 jobs arriving to the system since $\lambda a_1 \mu_{11}^{-1} > 1$. This requires some of the arriving class 1 jobs to be directed to non-specialist stations. If all class 2 jobs were routed to station 2 then $\rho_{22} = 0.985$ (i.e. the traffic intensity at station 2 due to class 2 jobs is close to 1). To minimise the total expected delay some class 2 traffic is routed to non-specialist stations. The amount of work routed to non-specialist stations is slightly greater under policy O. By considering the effective arrival rate of each job class to the system and appropriately weighting the optimal routing probabilities in Tables 2.1 and 2.2, policy F routes 6.26% of jobs to non-specialist stations whereas policy O routes 7.03% of jobs to non-specialist stations.

It is interesting to note that for the job classes whose specialist processing station is incapable of or not suited to processing all arriving jobs of that class then there exists a distinct station preference for the allocation of this overflow. The choice of the k_{gm} means that $\mu_{gm}^{-1} = \mu_{gn}^{-1}$ and $M_{gm} = M_{gn}$ when $|g - m| = |g - n|$, $m \neq n$, $g \in G$, m, $n \in \mathcal{M}$, i.e. stations m and n are equally capable of processing class g jobs. However, if $g > m$ then $p_{gm} = 0$. The rationale behind this routing policy is simple to understand. Consider stations 1 and 3 as non-specialist stations equally capable of handling class 2 jobs. Under the choices of the a_g the effective arrival rate of class 3 jobs is less than that of class 1 jobs. Additionally we have that $\mu_{gg}^{-1} = 0.5$ and $M_{gg} = 0.5$ for all g. Hence station 3 will have the greater available processing capability and delays would be reduced by routing class 2 to jobs to station 3.

It is obvious that such reasoning above applies more generally to Systems 1-6 under

consideration here. It must be noted that in all problems the amount of work routed to non-specialist stations is greater under policy O.

	Station 1	Station 2	Station 3	Station 4	Station 5
Class 1	0.8158		0.0256	0.0705	0.0881
Class 2		0.9705	0.0295		
Class 3			1.0000		
Class 4				1.0000	
Class 5					1.0000

Table 2.1: Optimal routing probabilities under policy F for System 2.

	Station 1	Station 2	Station 3	Station 4	Station 5
Class 1	0.8085	0.0082	0.0176	0.0744	0.0913
Class 2		0.9488	0.0512		
Class 3			1.0000		
Class 4				1.0000	
Class 5					1.0000

Table 2.2: Optimal routing probabilities under policy O for System 2.

(ii) The speed with which stations process jobs for which they are non-specialists depends critically upon the parameters k_{gm}. To examine the effect these parameters have upon the amount of work routed to non-specialist stations we allow k_{gm} to have the more general form

$$k_{gm} = \delta|g - m| + 1. \tag{2.42}$$

In (2.42) small values of δ suggest a greater capability of the stations to process jobs for which they are non-specialists. Large values of δ suggest a lesser capability to process such jobs.

Tables 2.3-2.8 display the total expected delay and the percentage of jobs routed to non-specialist stations under policies F and O as δ increases from 0.2 to 2.0 for Systems 1-6. In each problem considered λ is set to 95% of $\bar{\lambda}$. As δ increases

jobs take longer to process at non-specialist stations and there is less benefit in routing such jobs to these stations. The maximum amount of traffic feasible to the system will thus be greatly affected by the amount of work the specialist stations can process. Accordingly $\bar{\lambda}$ decreases as δ increases. With λ fixed at 95% of $\bar{\lambda}$, decreases in $\bar{\lambda}$ mean a smaller amount of jobs arriving at the system controller. Additionally, the first two moments of the service time distribution are independent of $\bar{\lambda}$. As δ increases the specialist stations (in particular the stations facing a comparably larger specialist job arrival rate) can process a greater percentage of their own traffic and the system controller will redirect less jobs to non-specialist stations. Jobs are then completed more quickly and the total expected delay decreases as δ increases.

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	7.41	7.04	73.54	15.88	67.09	17.21
0.4	7.06	6.71	72.32	13.08	62.47	14.40
0.6	6.82	6.48	71.64	11.00	59.94	12.20
0.8	6.64	6.31	71.26	9.52	58.13	10.47
1.0	6.49	6.17	70.91	8.79	55.35	9.10
1.2	6.37	6.06	70.64	8.17	53.05	8.47
1.4	6.27	5.96	70.41	7.63	51.37	7.95
1.6	6.19	5.88	70.21	7.16	50.08	7.50
1.8	6.11	5.81	70.03	6.74	49.05	7.09
2.0	6.05	5.75	69.85	6.36	48.21	6.73

Table 2.3: Percentage of jobs routed to non-specialist stations in System 1

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	9.30	8.84	92.14	11.80	81.68	12.78
0.4	8.91	8.46	90.84	9.77	75.44	10.52
0.6	8.64	8.21	90.18	8.32	71.91	9.00
0.8	8.44	8.02	89.74	7.18	68.86	7.86
1.0	8.29	7.88	89.47	6.26	66.67	7.03
1.2	8.17	7.76	89.28	5.51	65.29	6.32
1.4	8.07	7.67	89.15	5.11	64.40	5.70
1.6	7.99	7.59	89.72	4.49	63.65	5.15
1.8	7.91	7.52	89.57	4.21	61.94	4.68
2.0	7.84	7.45	89.40	3.97	60.42	4.32

Table 2.4: Percentage of jobs routed to non-specialist stations in System 2

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	10.66	10.13	108.48	16.04	96.47	17.80
0.4	10.00	9.50	106.26	13.00	89.17	14.67
0.6	9.58	9.10	105.12	11.15	83.68	11.66
0.8	9.28	8.82	104.46	9.73	80.03	10.29
1.0	9.06	8.60	104.05	8.61	77.78	9.18
1.2	8.88	8.44	103.77	7.69	76.46	8.26
1.4	8.74	8.30	103.95	6.92	74.46	7.49
1.6	8.61	8.18	103.86	6.24	72.86	6.88
1.8	8.51	8.09	103.81	5.65	71.86	6.35
2.0	8.42	8.00	103.78	5.14	71.22	5.86

Table 2.5: Percentage of jobs routed to non-specialist stations in System 3

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	12.56	11.93	127.30	13.43	110.24	14.62
0.4	11.86	11.27	125.35	10.37	101.24	11.41
0.6	11.43	10.86	124.55	8.52	95.61	9.39
0.8	11.13	10.57	124.04	7.43	91.24	7.93
1.0	10.90	10.35	123.80	6.59	88.35	7.09
1.2	10.71	10.18	123.68	5.88	85.92	6.40
1.4	10.56	10.03	123.51	5.29	84.14	5.84
1.6	10.44	9.92	123.34	4.79	82.99	5.36
1.8	10.33	9.82	123.18	4.36	82.23	4.94
2.0	10.25	9.73	124.04	4.00	81.71	4.57

Table 2.6: Percentage of jobs routed to non-specialist stations in System 4

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	13.73	13.04	142.59	16.05	124.70	17.39
0.4	12.75	12.12	139.45	12.34	114.79	13.34
0.6	12.17	11.56	138.02	10.22	107.31	11.00
0.8	11.78	11.19	137.37	8.54	103.54	9.38
1.0	11.48	10.91	137.00	7.59	99.14	8.05
1.2	11.25	10.68	136.63	6.82	95.75	7.27
1.4	11.06	10.50	136.33	6.18	93.57	6.64
1.6	10.90	10.36	136.10	5.64	92.04	6.10
1.8	10.77	10.23	136.29	5.16	90.67	5.64
2.0	10.66	10.13	136.11	4.74	89.50	5.24

Table 2.7: Percentage of jobs routed to non-specialist stations in System 5

			FCFS Scheduling		Optimal Scheduling	
δ	$\bar{\lambda}$	0.95 $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	% jobs to non-specialist stations	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	% jobs to non-specialist stations
0.2	15.63	14.85	161.77	12.99	137.66	14.27
0.4	14.64	13.91	159.22	9.93	125.58	10.67
0.6	14.05	13.35	158.18	8.12	118.29	8.76
0.8	13.65	12.97	157.93	6.79	113.52	7.36
1.0	13.35	12.68	157.71	5.76	110.03	6.50
1.2	13.12	12.46	158.15	5.14	107.33	5.70
1.4	12.92	12.28	158.14	4.65	104.70	5.08
1.6	12.76	12.13	158.00	4.23	102.56	4.67
1.8	12.63	12.00	157.76	3.87	101.09	4.33
2.0	12.51	11.89	157.46	3.55	100.15	4.02

Table 2.8: Percentage of jobs routed to non-specialist stations in System 6

(iii) Tables 2.9-2.14 display the results concluding our extended numerical study. We report the values of $\widehat{\Phi}^F(\mathbf{P}_F^*)$, $\widehat{\Phi}^O(\mathbf{P}_F^*)$, $\widehat{\Phi}^O(\mathbf{P}_O^*)$ and the relative improvement in the total expected delay gained via the implementation of policy O over policy F (columns 3,4,5 and 6, respectively), for a range of feasible λ. Note that $\widehat{\Phi}^O(\mathbf{P}_F^*)$ is the total expected delay under policy O using the optimal routing probabilities for policy F.

When the system experiences low traffic intensity, around 50% of $\bar{\lambda}$ say, the optimal total expected delay experienced under the two scheduling policies is similar. However, as reported in column 6, policy O shows an improvement over policy F as λ increases. Hence, while the choice of scheduling policy makes little difference to performance under low traffic intensities, scheduling traffic optimally at each station produces a marked reduction in the total expected delay in heavier traffic.

Finally, in comparing columns 4 and 5, we see that the optimal FCFS routing probabilities are relatively robust when stations schedule their work optimally.

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
3.25	50	3.46	3.45	3.45	0.25
3.57	55	4.30	4.27	4.26	0.87
3.90	60	5.34	5.27	5.25	1.80
4.22	65	6.69	6.52	6.48	3.06
4.55	70	8.48	8.14	8.08	4.69
4.87	75	10.98	10.35	10.24	6.75
5.19	80	14.73	13.56	13.36	9.33
5.52	85	20.98	18.71	18.36	12.52
5.84	90	33.47	28.61	27.95	16.49
6.17	95	70.91	56.80	55.35	21.95

Table 2.9: Total expected delay for varying λ in System 1

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
4.15	50	4.14	4.14	4.14	0.00
4.56	55	5.15	5.15	5.14	0.07
4.98	60	6.46	6.43	6.41	0.63
5.39	65	8.14	8.03	8.00	1.78
5.81	70	10.40	10.11	10.03	3.54
6.22	75	13.56	12.92	12.74	6.00
6.63	80	18.30	16.96	16.61	9.22
7.05	85	26.20	23.39	22.72	13.31
7.46	90	42.02	35.61	34.22	18.56
7.88	95	89.47	70.48	66.67	25.48

Table 2.10: Total expected delay for varying λ in System 2

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
4.53	50	4.76	4.76	4.76	0.01
4.98	55	5.97	5.96	5.96	0.27
5.43	60	7.51	7.45	7.43	1.06
5.89	65	9.49	9.30	9.26	2.43
6.34	70	12.12	11.70	11.60	4.33
6.79	75	15.81	14.93	14.73	6.81
7.24	80	21.33	19.58	19.21	9.97
7.70	85	30.53	26.99	26.28	13.94
8.15	90	48.92	41.13	39.66	18.93
8.60	95	104.05	81.70	77.78	25.24

Table 2.11: Total expected delay for varying λ in System 3

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
5.45	50	5.40	5.40	5.40	0.00
5.99	55	6.71	6.71	6.71	0.00
6.54	60	8.47	8.46	8.45	0.26
7.08	65	10.79	10.69	10.66	1.23
7.63	70	13.91	13.59	13.47	3.16
8.17	75	18.30	17.45	17.20	5.99
8.71	80	24.89	22.96	22.47	9.73
9.26	85	35.88	31.63	30.67	14.51
9.81	90	57.87	47.92	45.92	20.64
10.35	95	123.80	93.61	88.35	28.63

Table 2.12: Total expected delay for varying λ in System 4

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
5.73	50	5.90	5.90	5.90	0.00
6.31	55	7.42	7.42	7.42	0.06
6.89	60	9.40	9.37	9.36	0.46
7.46	65	12.00	11.86	11.80	1.62
8.04	70	15.48	15.04	14.92	3.61
8.61	75	20.35	19.31	19.05	6.38
9.18	80	27.64	25.41	24.88	9.98
9.76	85	39.80	35.03	34.02	14.51
10.33	90	64.11	53.20	51.15	20.21
10.91	95	137.00	104.35	99.14	27.64

Table 2.13: Total expected delay for varying λ in System 5

λ	% of $\bar{\lambda}$	$\widehat{\Phi}^F(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$	$100\left(\frac{\widehat{\Phi}^F(\mathbf{P}_F^*)-\widehat{\Phi}^O(\mathbf{P}_O^*)}{\widehat{\Phi}^F(\mathbf{P}_F^*)}\right)$
6.68	50	6.55	6.55	6.55	0.00
7.34	55	8.12	8.12	8.12	0.00
8.01	60	10.25	10.25	10.24	0.09
8.68	65	13.15	13.09	13.06	0.70
9.35	70	17.13	16.82	16.07	2.49
10.01	75	22.74	21.79	21.50	7.40
10.68	80	31.16	28.84	28.19	9.52
11.35	85	45.22	39.86	38.51	14.84
12.02	90	73.33	60.50	57.51	21.68
12.68	95	157.71	118.69	110.03	30.24

Table 2.14: Total expected delay for varying λ in System 6

We now adapt our study by introducing further controls to the system set up. Here, we consider situations in which, given a fixed budget, the system controller can only employ a limited number of stations with which to process the arriving jobs but is allowed total freedom in deciding the exact configuration of the specialist stations to be used. The problem for the system controller is to choose the optimal mix of specialist stations under the budget constraints. If so desired, the system controller can decide to have all available stations to have the same job class specialisation. Considering that the job arrival stream consists of multiple jobs classes one would expect such a set up to be appropriate only in cases where a single job class dominates the arrival stream. In what follows we shall consider a modified version of System 2 in which the number of stations available for processing, M, comprises 3, 4 and 5 stations. As in the original System 2, $|G| = 5$, i.e. there are 5 job classes. We use m_σ, $m \in \mathcal{M}$, $\sigma \in G$ to denote each station in the network and its job class specialisation. With this notation the following two station configurations $(1_1, 2_1, 3_1, 4_1, 5_2)$ and $(1_1, 2_1, 3_1, 4_2, 5_1)$ are identical in their mix of expertise and only differ in the given labelling (i.e. the job class specialisation identifiers for stations 4 and 5 have interchanged). We need only consider *distinct* station configurations (i.e. a unique mix of expertise). Hence, for systems comprising 3, 4 and 5 stations there exist 35, 70 and 126 distinct station configurations respectively. We retain the gamma service time distribution with scale parameter α_{m_σ} and shape parameter $k_{gm_\sigma}\beta_{m_\sigma}$ where the general form of the k_{gm_σ} as given by (2.42) is replaced by

$$k_{gm_\sigma} = \delta|g - \sigma| + 1,$$

which appropriately takes into account the specialisation of the station. As before, we set $\alpha_{m_\sigma} = 2$, $\beta_{m_\sigma} = 1$, $\delta = 1.0$ and the delay weights, c_g, to be equal to 1. It now remains to specify the job class arrival probabilities. These probabilities will have a marked effect on the station configuration. For example, a system which has a comparatively large effective arrival rate in one job class requires a suitable station configuration that deals with the demand placed upon the system by this job class while still providing processing

capacity for the remaining set of job classes. However, there will be less need for stations specialist in processing the jobs classes with a low effective arrival rate to the system. To take account of this we shall consider two sets of arrival probabilities. We shall use the original set of arrival probabilities, $a_g = 0.35 - 0.05g,\ g \in G$, in which job class 1 has the largest effective arrival rate and job class 5 the lowest. Our second set of arrival probabilities, $a'_g = 1/M,\ g \in G$, has the same effective arrival rate for all job classes.

$M=5$		$M=4$		$M=3$	
$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma, 5_\sigma)$	$\bar{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma)$	$\bar{\lambda}$	$(1_\sigma, 2_\sigma; 3_\sigma)$	$\bar{\lambda}$
(1,2,3,4,5)	8.29	(1,2,3,4)	6.97	(1,2,4)	4.44
(1,1,2,3,4)	8.13	(1,2,3,5)	6.86	(1,3,4)	4.32
(1,2,2,3,4)	8.07	(1,2,4,5)	6.25	(1,2,5)	4.18
(1,2,3,4,4)	8.05	(1,2,4,4)	6.11	(1,3,5)	4.17
⋮	⋮	⋮	⋮	⋮	⋮
(4,4,4,5,5)	3.57	(4,4,4,5)	2.95	(4,4,4)	2.22
(4,4,5,5,5)	3.40	(4,4,5,5)	2.79	(4,4,5)	2.17
(4,5,5,5,5)	3.20	(4,5,5,5)	2.60	(4,5,5)	2.00
(5,5,5,5,5)	2.86	(5,5,5,5)	2.29	(5,5,5)	1.71

Table 2.15: Maximum feasible arrival rates, $\bar{\lambda}$, for a range of station configurations in which the set of arrival probabilities is given by $a_g,\ g \in G$.

$M=5$		$M=4$		$M=3$	
$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma, 5_\sigma)$	$\bar{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma)$	$\bar{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma)$	$\bar{\lambda}$
(1,2,3,4,5)	10.00	(1,2,4,5)	6.25	(1,3,5)	4.28
(1,2,2,4,5)	8.13	(1,2,3,5)	6.13	(1,3,4)	4.12
(1,2,4,4,5)	8.13	(1,3,4,5)	6.13	(2,3,5)	4.12
(1,1,3,4,5)	8.06	(1,3,3,5)	5.71	(2,3,4)	4.00
⋮	⋮	⋮	⋮	⋮	⋮
(1,1,1,1,2)	3.79	(1,1,1,2)	3.10	(1,1,2)	2.38
(4,5,5,5,5)	3.79	(4,5,5,5)	3.10	(4,5,5)	2.38
(1,1,1,1,1)	3.33	(1,1,1,1)	2.67	(1,1,1)	2.00
(5,5,5,5,5)	3.33	(5,5,5,5)	2.67	(5,5,5)	2.00

Table 2.16: Maximum feasible arrival rates, $\bar{\lambda}$, for a range of station configurations in which the set of arrival probabilities is given by $a'_g,\ g \in G$.

To proceed we need to determine the maximum (total) system arrival rate, $\bar{\lambda}$, for each problem. This is achieved by solving a suitably amended version of the optimisation

problem in (2.39) taking into account the specialisation of each station. In Tables 2.15 and 2.16 we report a sample of station configurations (here, we simply provide the job class specialisation, σ, of each station) and their maximal arrival rate for the two choices of system arrival probabilities, a_g and a'_g, respectively. For each network size, $M = 3, 4, 5$, we report only the four station configurations with the largest maximum (total) system arrival rate and the four configurations with the lowest. Given the system arrival probabilities a_g for the problems considered in Table 2.15, jobs from class 1 arrive at the system with the greatest frequency and jobs from class 5 with the lowest frequency. It can be seen that station configurations favouring specialisation in the job classes with the greater effective system arrival rate, while still providing enough processing capacity for the job classes with lower effective system arrival rates, can allow a greater amount of traffic to access the system. Station configurations utilising only stations specialising in the job classes with lower effective arrival rates can only cope with light traffic loads. We can compare these station configurations with those in Table 2.16. Under the system arrival probabilities a'_g all job classes arrive at the system with the same rates. Here configurations that essentially consist of a balance of station specialisations, i.e. that in some way try to uniformly supply processing potential for the entire job arrival stream, can handle the greatest amount of traffic. Configurations comprising mainly of stations best equipped to process only a small range of the arriving job classes can only cope with light traffic loads.

We must note that, under a'_g and the assumptions about the service times, we experience symmetries between station configurations. For example, station configurations whose mix of expertise given by (1,1,1,1,1) and (5,5,5,5,5) have an identical (total) system arrival rate ($\bar{\lambda} = 3.33$). Both these systems achieve the same (overall) system performance, which is the primary concern here. However, while not of concern in the current problem, it is clear that jobs from class 1 (resp. 5) will experience greater delays in system (5,5,5,5,5) (resp. (1,1,1,1,1)). In reporting our results from the numerical study we shall include all the results for the distinct server configurations, including the configurations

achieving the same system performance.

We are now in a position to investigate how the different station configurations perform under the two scheduling policies F and O. Naturally our interest lies in station configurations that we would expect to provide the best service to the arriving job classes. From Tables 2.15-2.16 an obvious indicator to such configurations is the maximum amount of traffic allowable to the system. In what follows, we use $\hat{\lambda}$ to denote the maximum (total) system arrival rate over all station configurations.

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma, 5_\sigma)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$
7.88	95	(1,2,3,4,5)	89.47	66.67
7.46	90	(1,2,3,4,5)	42.02	34.22
		(1,1,2,3,4)	44.00	31.99
		(1,2,2,3,4)	45.49	35.31
		(1,2,3,4,4)	46.03	40.23
		(1,2,3,3,5)	49.66	45.21
		(1,2,3,3,4)	50.44	45.61
		(1,2,2,3,5)	50.68	41.97
		(1,1,2,3,5)	54.48	40.83
7.05	85	(1,2,3,4,5)	26.20	22.72
		(1,1,2,3,4)	25.06	19.78
		(1,2,2,3,4)	25.12	20.96
		(1,2,3,4,4)	25.07	23.16
		(1,2,3,3,5)	26.05	24.50
		(1,2,3,3,4)	26.54	24.86
		(1,2,2,3,5)	26.57	23.24
		(1,1,2,3,5)	28.34	22.87
		(1,2,3,5,5)	33.18	29.14
		(1,2,2,4,5)	36.17	31.29
		(1,2,4,4,5)	51.02	42.87
		(1,2,2,4,4)	40.73	38.70
		(1,1,2,4,5)	56.79	42.08

Table 2.17: Total expected delay over a range of station configurations for varying λ in which $M = 5$ and the set of arrival probabilities is given by $a_g,\ g \in G$

In Tables 2.17-2.19 we report the total expected delay under policies F and O for $M = 5, 4, 3$, respectively, over a range of λ values in which the arrival probabilities are given by $a_g,\ g \in G$. For each λ used (ranging from 85-95% of $\hat{\lambda}$) we only consider station configurations for which 95% of the maximum feasible (total) arrival rate for that

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$
6.62	95	(1,2,3,4)	74.12	53.20
6.28	90	(1,2,3,4)	35.03	27.63
		(1,2,3,5)	41.28	33.33
5.93	85	(1,2,3,4)	22.00	18.42
		(1,2,3,5)	24.40	20.78
		(1,2,4,5)	69.84	47.92

Table 2.18: Total expected delay over a range of station configurations for varying λ in which $M = 4$ and the set of arrival probabilities is given by $a_g,\ g \in G$

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma,)$	$\widehat{\Phi}^O(\mathbf{P}_F^*)$	$\widehat{\Phi}^O(\mathbf{P}_O^*)$
4.42	95	(1,2,4)	53.93	36.32
4.00	90	(1,2,4)	25.58	18.64
		(1,3,4)	34.41	24.33
3.78	85	(1,2,4)	16.12	12.50
		(1,3,4)	19.35	14.63
		(1,2,5)	26.45	18.46
		(1,3,5)	26.44	19.31
		(1,2,3)	47.59	26.18

Table 2.19: Total expected delay over a range of station configurations for varying λ in which $M = 3$ and the set of arrival probabilities is given by $a_g,\ g \in G$

configuration, $\bar{\lambda}$, is greater than or equal to λ. For each problem the total expected delay is greater under policy F than under policy O. However, it is interesting to note that, in Table 2.17, station configurations that have a greater maximum total arrival rate $\bar{\lambda}$ may not necessarily be the best combination of stations to utilise. Consider the station configurations (1,2,3,4,5) and (1,1,2,3,4). The first configuration has a specialist processing station for each job class. For this configuration $\hat{\lambda} = \bar{\lambda} = 8.29$. The second configuration provides more specialist processing capability for job class 1 and no specialist processing provision for job class 5. Here, $\hat{\lambda} > \bar{\lambda} = 8.13$. Under policy O, the total expected delay for configuration (1,1,2,3,4) is less than that for configuration (1,2,3,4,5) when λ is set to 85 and 90% of $\hat{\lambda}$. It is similarly the case under policy F when λ is set to 85% of $\hat{\lambda}$. As we decrease λ the spare processing capacity increases across the network and routing decisions can take advantage of the spare capacity available for processing job classes with higher effective arrival rates. Obviously, when stations schedule their work optimally the system controller can take greater advantage of the informed station configuration. In Tables 2.18 and 2.19 the reduction in the number of available stations means that the system controller has less processing capacitity available and jobs will have to be routed to non-specialist stations. Here, the station configuration for which $\bar{\lambda} = \hat{\lambda}$ attains the minimum total expected delay for all λ values considered.

In Tables 2.20-2.22 we report the total expected delay under policies F and O for $M = 5, 4, 3$, respectively, over a range of λ values in which the arrival probabilities are given by a'_g, $g \in G$. Again, for each λ used (ranging from 85-95% of $\hat{\lambda}$) we only consider station configurations for which 95% of the maximum feasible (total) arrival rate for that configuration, $\bar{\lambda}$, is greater than or equal to λ. When $M = 5$ we see from Table 2.20 that the total expected delay experienced under both policies is the same. For these problems, each job class arrives to the system at the same rate and for configuration (1,2,3,4,5) each station is able to process the full arrival stream of its specialist job class. Hence, no routing of jobs to non-specialist stations is required and the two scheduling policies at each station have identical performance. As we restrict the number of stations

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma, 5_\sigma)$	$\widehat{\Phi}^O(\mathbf{P}^*_F)$	$\widehat{\Phi}^O(\mathbf{P}^*_O)$
9.50	95	(1,2,3,4,5)	95.00	95.00
9.00	90	(1,2,3,4,5)	45.00	45.00
8.50	85	(1,2,3,4,5)	28.33	28.33

Table 2.20: Total expected delay over a range of station configurations for varying λ in which $M = 5$ and the set of arrival probabilities is given by $a'_g,\ g \in G$

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma, 4_\sigma)$	$\widehat{\Phi}^O(\mathbf{P}^*_F)$	$\widehat{\Phi}^O(\mathbf{P}^*_O)$
5.94	95	(1,2,4,5)	74.77	45.90
5.62	90	(1,2,4,5)	35.39	24.09
		(1,3,4,5)	43.96	28.46
		(1,2,3,5)	43.96	28.46
5.31	85	(1,2,4,5)	22.27	16.40
		(1,3,4,5)	25.55	18.17
		(1,2,3,5)	25.55	18.17
		(1,3,3,5)	34.41	27.43
		(2,3,4,5)	70.24	38.75
		(1,2,3,4)	70.24	38.75

Table 2.21: Total expected delay over a range of station configurations for varying λ in which $M = 4$ and the set of arrival probabilities is given by $a'_g,\ g \in G$

available for processing, jobs will have to be routed to non-specialist stations. In such circumstances the station configuration for which $\bar{\lambda} = \hat{\lambda}$ has best possible performance amongst all station configurations for the values of λ considered here. For these problems the total expected delay under policy O is less than that under policy F.

The problem of determining the optimal mix of expertise is a two stage process. The first stage requires the determination of the maximum (total) system arrival rate for each station configuration. The second stage involves the calculation of the total expected delay for a given system arrival rate λ for each stable station configuration (i.e. a steady state solution exists with finite queue lengths). The optimal mix of expertise is given by the station configuration achieving the minimum total expected delay over all stable station configurations. This process of determining the optimal mix of expertise is computationally demanding. A good heuristic for the choice of station configuration is to select the station configuration which achieves $\hat{\lambda}$, the maximum (total) system arrival rate over

λ used	% of $\hat{\lambda}$	$(1_\sigma, 2_\sigma, 3_\sigma,)$	$\widehat{\Phi}^O(\mathbf{P}^*_F)$	$\widehat{\Phi}^O(\mathbf{P}^*_O)$
4.07	95	(1,3,5)	52.58	38.92
3.86	90	(1,3,5)	25.02	19.41
		(2,3,5)	40.88	28.83
		(1,3,4)	40.88	28.83
3.64	85	(1,3,5)	15.82	12.78
		(2,3,5)	21.25	15.96
		(1,3,4)	21.25	15.96
		(2,3,4)	28.23	20.94
		(1,2,5)	37.18	25.71
		(1,4,5)	37.18	25.71
		(2,4,5)	49.23	32.13
		(1,2,4)	49.23	32.13

Table 2.22: Total expected delay over a range of station configurations for varying λ in which $M = 3$ and the set of arrival probabilities is given by $a'_g,\ g \in G$

all station configurations. From the results of the numerical study this station configuration may not be the optimal mix of expertise. However, for the problems considered this configuration always performs well and has the advantage of being able to cope with the greatest amount of traffic. Additionally, application of the heuristic significantly reduces the amount computational effort, requiring only the determination of the mix of expertise achieving $\hat{\lambda}$.

2.10 Network Design

Until now, our routing problem has been concerned in the main with some policy for directing incoming traffic to a set of stations, the number of which is known. It was only in the numerical study of the last section where we considered some possible implications of the construction of the system network to the solution of the routing problem of interest. We now consider further the important question of network design. The system controller could, for example, have a brief to construct a network in order to achieve some predetermined performance objective. The type of routing policy used to attain this goal will be naturally dependent upon the sophistication of the network system available.

This still leaves open a more demanding question: how many stations (and, possibly, of what kind) should be employed to optimise the overall network performance? For fixed and known arrival rates it is clear that the addition of further machines to the present network will lead to a reduction in the overall costs incurred by that network. However as the number of competing jobs can never fall below zero, and with no restriction on the number of machines available, there must be some point at which the cost of adding an extra machine (with some purchase/implementation cost) will outweigh any reward gained.

We find few results relating to this problem in the literature. Weber (1980) considers the optimal allocation of a fixed number of servers to a system of $G/GI/\eta$ queues in order to minimise the overall mean queueing time. At each queue the servers operate in parallel and process jobs according to a FCFS scheduling policy. He is able to show that the mean queueing time in a $G/GI/\eta$ queue is a non-increasing and convex function of the number of servers, η. This corresponds to a law of diminishing returns from adding additional servers. Dacre (1999) considers the effect that the system composition has on the overall cost under an optimal static routing regime incorporating optimal local scheduling. He shows that the system cost is a non-increasing, supermodular function of the set of stations that make up the system. In full generality this result is dependent upon a conjecture regarding properties of supermodular functions. However, the conjecture is not required to establish the result in two special cases where there is a single generic job class and where the stations are homogeneous in a fully convex system.

Here, we focus on the special case where stations are homogeneous. The main result of this section states that when the returns at each station are NE-convex additional stations contribute diminishing returns under the static routing policy ESP. We conclude this section with a computational study that examines the nature of solutions of a simple design problem under the static ESP regime.

2.10.1 Constructing Networks of Homogeneous Stations

The difficulties posed by the network design problem constitute a considerable challenge to analysis. Our broad approach is still that of the static routing problem proposed at the beginning of this chapter. Hence, we can allow for multiple job types belonging to either the dedicated or generic job classes. When stations are homogeneous we assume that job processing requirements are station independent and that each station attracts (stochastically) identical dedicated traffic. Our objective remains to route the incoming generic traffic across the stations and to schedule the work at each station to minimise the long-run average holding cost rate as given by (2.1). In particular, our aim is to analyse the performance of such policies as the number of stations comprising the network increases (alternatively decreases).

In Section 2.8.2 we considered the special case of static routing when stations are homogeneous. In such circumstances the ESP routing policy, $p_{gm} = 1/M$, $g \in G$, $m \in \mathcal{M}$, in which the load is divided equally between the stations would appear to be the obvious candidate for the routing policy. Indeed, from Theorem 20, ESP is the optimal static routing policy when the optimal station return, Ψ, at each station is convex. However, in the more general case in which the optimal station returns are assumed to be NE-convex only (see Definition 8 and Theorem 14) ESP is generally suboptimal. Numerical investigations by Dacre (1999) imply that ESP performs well for problems close to the conditions which yield Theorem 21 and was able to provide bounds for its performance in a number of special cases. These considerations and the simple structure of ESP lead us to consider the implications of ESP routing as the number of stations comprising the network changes.

In the following result we establish that additional stations contribute diminishing returns under ESP . We use Φ^{ESP} in Theorem 23 to denote the system cost under ESP, given by

$$\Phi^{ESP}_{\{1,\ldots,M\}}(\boldsymbol{\lambda}^G) = M\Psi\left(\frac{\boldsymbol{\lambda}^G}{M}\right).$$

Theorem 23

When stations are homogeneous and the optimal return Ψ at each station is NE-convex then

(a) $\Phi^{ESP}_{\{1,\ldots,M\}}(\boldsymbol{\lambda}^G)$ *is decreasing in* M,

(b) $\Phi^{ESP}_{\{1,\ldots,M\}}(\boldsymbol{\lambda}^G) - \Phi^{ESP}_{\{1,\ldots,M+1\}}(\boldsymbol{\lambda}^G)$ *is decreasing in* M.

Proof

From Definition 8 and Theorem 14 we have that, for any feasible system arrival rate $\boldsymbol{\lambda}^G$,

$$\frac{M}{M+1}\Psi\left(\frac{\boldsymbol{\lambda}^G}{M}\right) + \frac{1}{M+1}\Psi(\mathbf{0}) \geq \Psi\left(\frac{\boldsymbol{\lambda}^G}{M+1}\right),$$

and hence,

$$M\Psi\left(\frac{\boldsymbol{\lambda}^G}{M}\right) \geq (M+1)\Psi\left(\frac{\boldsymbol{\lambda}^G}{M+1}\right). \tag{2.43}$$

The quantities in (2.43) are plainly the costs associated with homogeneous systems comprising M and $M+1$ stations under ESP. We have established (a).

Similarly, we infer

$$\frac{M}{2(M+1)}\Psi\left(\frac{\boldsymbol{\lambda}^G}{M}\right) + \frac{M+2}{2(M+1)}\Psi\left(\frac{\boldsymbol{\lambda}^G}{M+2}\right) \geq \Psi\left(\frac{\boldsymbol{\lambda}^G}{M+1}\right),$$

and hence,

$$M\Psi\left(\frac{\boldsymbol{\lambda}^G}{M}\right) - (M+1)\Psi\left(\frac{\boldsymbol{\lambda}^G}{M+1}\right) \geq (M+1)\Psi\left(\frac{\boldsymbol{\lambda}^G}{M+1}\right) - (M+2)\Psi\left(\frac{\boldsymbol{\lambda}^G}{M+2}\right)$$

and we have (b). □

2.10.2 Numerical Study

We now present the results of a computational study of a simple design problem. The system controller has to construct a network of homogeneous stations in order to minimise the long-run average system costs under the ESP routing policy. The number of stations, M, comprising the network remains to be determined. Each station m has an associated expected running cost per unit time, denoted K, independent of the amount of work received. This cost could include the purchase/implementation and maintenance costs over the expected lifetime of the station. As we only consider homogeneous stations we may safely assume identical running costs for each station m. The system arrival rates of all job classes, the processing capabilities of each station and the cost characteristics of the queueing job classes at each station are considered fixed and known. The overall system cost (holding costs plus running costs) under ESP, denoted Ω, is given by

$$\Omega(M) = \Phi^{ESP}_{\{1,\ldots,M\}}(\boldsymbol{\lambda}^G) + MK. \tag{2.44}$$

The problem of optimal system design under ESP routing can be expressed by the minimisation problem

$$\min_M : \Omega(M). \tag{2.45}$$

The function $\Omega : \mathbb{N} \to \mathbb{R}_+$ in (2.44) is the sum of two convex functions and is itself convex. The optimal solution to (2.45) is, therefore, to select M stations where $\Omega(M-1) \geq \Omega(M) \leq \Omega(M+1)$.

In the study we suppose that the stations are modelled as preemptive two-class $M/M/1$ systems with local scheduling according to the $c\mu$-rule. We shall initially consider the special case in which the job classes are stochastically indistinguishable and take $\mu_1 = \mu_2 = 1$ for the exponential service rates of the two job classes. Theorem 20 applies in this case, hence, ESP is the optimal static routing policy. Under the $c\mu$-rule higher priority is accorded to the class with the larger holding cost rate. We fix $c_2 = 1$ and take $c_1 \geq 1$

ensuring that class 1 always has higher priority. The optimal number of stations and the overall system costs under ESP were computed for problems with c_1 ranging from 1.5 to 6, K ranging from 0.25 to 2 and $\rho = \lambda_1 + \lambda_2$ taking the values, 1.0, 1.3, 1.6 and 1.9. The ρ-values reflect situations in which the traffic intensity ranges from medium to heavy traffic for the minimum possible system set-up consisting of two stations. For each ρ-value, (λ_1, λ_2) pairs were randomly generated by sampling λ_2 from a $U[0.2, \rho - 0.2]$ distribution then choosing $\lambda_1 = \rho - \lambda_2$. A simple closed form expression for the long-run average holding cost rate for this system under ESP, $\Omega(M)$, is available. Therefore, it is entirely possible to generate a large number of results in a relatively short time. However, in the study we shall only present a small selection of results for each problem set that are typical of our findings. Our main interest lies in how the optimal decision is affected by changes to the system parameters. Under the varying traffic intensities we consider how the optimal decision changes as the differences between the competing job classes becomes more pronounced and as the expected running costs of the stations increases.

(ρ, c_1)	λ_1	λ_2	$\Omega(M)$	M
1.0,1.5	0.598	0.402	2.623	3
1.0,3.0	0.598	0.402	3.740	3
1.0,6.0	0.598	0.402	5.850	4
1.3,1.5	0.797	0.503	3.424	4
1.3,3.0	0.797	0.503	4.903	5
1.3,6.0	0.797	0.503	7.748	5
1.6,1.5	0.996	0.604	4.225	5
1.6,3.0	0.996	0.604	6.070	6
1.6,6.0	0.996	0.604	9.631	7
1.9,1.5	1.195	0.705	5.027	6
1.9,3.0	1.195	0.705	7.240	7
1.9,6.0	1.195	0.705	11.517	8

Table 2.23: Optimal number of stations and overall system costs for the ESP routing policy for increasing c_1. Stations are homogeneous with identical running costs and stochastically indistinguishable job classes. $K = 0.25,\ c_2 = 1$.

When the jobs are stochastically indistinguishable we can only introduce differences between the job classes by altering the holding cost rates. In Table 2.23 we report the

			$c_1 = 1.5$		$c_1 = 3.0$		$c_1 = 6.0$	
ρ	λ_1	λ_2	$\Omega(M)$	M	$\Omega(M)$	M	$\Omega(M)$	M
1.0	0.233	0.767	2.376	3	2.754	3	3.511	3
	0.500	0.500	2.550	3	3.449	3	5.188	4
	0.706	0.294	2.712	3	2.754	4	6.612	5

Table 2.24: Optimal number of stations and overall system costs for the ESP routing policy for a range of arrival rates. Stations are homogeneous with identical running costs and stochastically indistinguishable job classes for a range of arrival rates. $K = 0.25,\ c_2 = 1.$

optimal decision as c_1 increases. We see that as c_1 increases it is optimal to build more stations into the network. Obviously, the greater the traffic intensity the greater the requirement for more stations in an optimal network. We also find that the optimal decision is sensitive to the arrival rates of the two job classes as they become more dissimilar. In Table 2.24 we see that as c_1 increases more stations are required in situations in which there are a greater number of arrivals from the job class with the larger holding cost rate.

(ρ, K)	λ_1	λ_2	$\Omega(M)$	M
1.0,0.25	0.598	0.402	2.624	3
1.0,0.5	0.598	0.402	3.373	3
1.0,1.0	0.598	0.402	4.427	2
1.0,2.0	0.598	0.402	6.427	2
1.3,0.25	0.797	0.503	3.424	4
1.3,0.5	0.797	0.503	4.337	3
1.3,1.0	0.797	0.503	5.837	3
1.3,2.0	0.797	0.503	8.377	2
1.6,0.25	0.996	0.604	4.225	5
1.6,0.5	0.996	0.604	5.330	4
1.6,1.0	0.996	0.604	7.174	3
1.6,2.0	0.996	0.604	10.174	3
1.9,0.25	1.195	0.705	5.027	6
1.9,0.5	1.195	0.705	6.350	5
1.9,1.0	1.195	0.705	8.471	4
1.9,2.0	1.195	0.705	12.175	3

Table 2.25: Optimal number of stations and overall system cost for the ESP routing policy for increasing K. Stations are homogeneous stations with identical running costs and stochastically indistinguishable job classes. $c_1 = 1.5,\ c_2 = 1.$

When the cost of each station grows we would expect that the system architect will wish to introduce fewer stations to the network. In Table 2.25 we report how increasing station costs, K, affects the optimal decision. We see that as K increases fewer stations are employed in the optimal system design. As previously, increasing ρ causes an increase in the optimal number of stations.

We now extend the study to consider scenarios in which the job classes are stochastically distinct. The reader should note that, for such problems ESP is no longer optimal. However, from Theorem 23 we have that, $\Phi_{\mathcal{M}}^{ESP}$, the long-run average holding cost rate for the system under ESP is a non-increasing, convex function of the number of stations comprising the network. Thus, overall system costs (including holding costs and running costs) given by (2.44) are convex in the number of stations. Our study now considers problems in which the job classes may, additionally, differ in the completion rates of their job processing times.

Again, we shall suppose that the stations are modelled as preemptive two-class $M/M/1$ systems with local scheduling according to the $c\mu$-rule. We take $c_2 = 1$ and $\mu_2 = 1$ and allow the class 1 parameters c_1 and μ_1 to take values such that $c_1\mu_1 \geq c_2\mu_2$, ensuring that the $c\mu$-rule gives higher priority to class 1 jobs. The optimal number of stations and the overall system costs under ESP were computed for problems taking a range of values for the parameters c_1, μ_1 and K. For each ρ-value (again, taking the values 1.0, 1.3, 1.6 and 1.9) (λ_1, λ_2) pairs were generated randomly by sampling λ_2 from a $U[0.2, \rho - 0.2]$ distribution then choosing $\lambda_1 = \mu_1(\rho - \lambda_2)$.

In Table 2.26 we report the optimal decision as both c_1 and μ_1 increase. We see that, as in the stochastically indistinguishable case, increases in ρ and c_1 lead to more stations in an optimal network. For the cases where $c_1 = 1$ the job classes only differ in their respective processing requirements. Here, increases in μ_1 lead to fewer stations being employed in an optimal network. This remains the case as the holding costs of class 1 jobs increase.

In Table 2.27 we report the optimal decision as both K and μ_1 increase. As previously,

			$c_1 = 1.0$		$c_1 = 1.5$		$c_1 = 3.0$		$c_1 = 6.0$	
(ρ, μ_1)	λ_1	λ_2	$\Omega(M)$	M	$\Omega(M)$	M	$\Omega(M)$	M	$\Omega(M)$	M
1.0,1.25	0.624	0.500	2.100	3	2.340	3	3.059	3	4.483	4
1.0,1.5	0.749	0.500	2.000	3	2.200	3	2.799	3	3.998	3
1.0,2.0	0.999	0.500	1.875	3	2.025	3	2.475	3	3.374	3
1.3,1.25	0.812	0.651	2.734	4	3.044	4	3.974	4	5.816	5
1.3,1.5	0.974	0.651	2.605	4	2.864	4	3.639	4	5.189	4
1.3,2.0	1.299	0.651	2.445	4	2.639	4	3.220	4	4.383	4
1.6,1.25	0.999	0.801	3.368	5	3.748	5	4.890	5	7.151	6
1.6,1.5	1.199	0.801	3.211	5	3.528	5	4.479	5	6.381	5
1.6,2.0	1.598	0.801	3.001	4	3.250	4	3.966	5	5.393	5
1.9,1.25	1.186	0.951	4.003	6	4.454	6	5.806	6	8.488	7
1.9,1.5	1.423	0.951	3.804	5	4.193	6	5.320	6	7.575	6
1.9,2.0	1.898	0.951	3.549	5	3.842	5	4.713	6	6.404	6

Table 2.26: Optimal number of stations and overall system cost for the ESP routing policy for increasing c_1 and μ_1. Stations are homogeneous with identical running costs and stochastically distinct job classes. $K = 0.25$, $c_2 = 1$.

as K increases less stations are employed in the optimal design. Again, the requirement is for more stations as the traffic intensity grows. The effect of μ_1 is harder to discern for the problems considered. Any increase in μ_1 corresponds to a decrease in the overall system cost. For small station costs ($K = 0.25$) increases in the class 1 processing rates cause a reduction in optimal number of stations. This is clearly the case in Table 2.26 in which all stations costs, K, are fixed at 0.25.

Our computations highlight the behaviour of the optimal decision with respect to parameter changes within the network. As the traffic intensity grows it is obvious that additional stations are required to cope with the increased demand. Similarly, if one job class is more expensive in relation to another, extra stations will reduce queue lengths across the network. As the processing power of the stations increase the network is able to cope with a greater demand for service. However, all such factors are constrained by the cost of the individual stations. A system architect will have greater freedom in constructing a suitable network when station costs are low. His options are more limited as the station cost becomes large.

			$K = 0.25$		$K = 0.5$		$K = 1.0$		$K = 2.0$	
(ρ, μ_1)	λ_1	λ_2	$\Omega(M)$	M	$\Omega(M)$	M	$\Omega(M)$	M	$\Omega(M)$	M
1.0,1.25	0.624	0.500	2.100	3	2.800	2	3.800	2	5.800	2
1.0,1.5	0.749	0.500	2.000	3	2.667	2	3.667	2	5.667	2
1.0,2.0	0.999	0.500	1.875	3	2.500	2	3.500	2	5.500	2
1.3,1.25	0.812	0.651	2.734	4	3.565	3	5.065	3	7.343	2
1.3,1.5	0.974	0.651	2.605	4	3.412	3	4.912	3	7.096	2
1.3,2.0	1.299	0.651	2.445	4	3.221	3	4.721	3	6.787	2
1.6,1.25	0.999	0.801	3.368	5	4.400	4	6.086	3	9.086	3
1.6,1.5	1.199	0.801	3.211	5	4.223	4	5.858	3	8.858	3
1.6,2.0	1.598	0.801	3.001	4	4.001	4	5.572	3	8.572	3
1.9,1.25	1.186	0.951	4.003	6	5.258	4	7.258	4	10.664	3
1.9,1.5	1.423	0.951	3.804	5	5.017	4	7.017	4	10.319	3
1.9,2.0	1.898	0.951	3.549	5	4.715	4	6.715	4	9.888	3

Table 2.27: Optimal number of stations and overall system cost for the ESP routing policy for increasing K and μ_1. Stations are homogeneous with identical running costs and stochastically distinct job classes. $c_1 = c_2 = 1$.

2.11 Conclusion

The static routing model described in Section 2.2 exhibits many desirable features that are of real concern in many contemporary applications. We have introduced and discussed the application of the achievable region approach to static routing problems in which the stations comprising the network schedule their work optimally. In Section 2.9 we applied the methods of Sections 2.3-2.8 for developing optimal static routing polices to a problem considered by Becker *et al.* (2000) in which a multi-class job population seeks service from a network of heterogeneous stations modelled as $M/G/1$ queueing systems. In the numerical study we showed that incorporating optimal local scheduling into the routing model can lead to significant improvements over a standard FCFS regime. This improvement was most noticeable under heavy traffic loads. In an adaptation of this problem we described a heuristic policy to determine the optimal mix of expertise to be built into the system design under a limited budget constraint. The policy, which selects the station configuration that is able to cope with the greatest traffic load over all configurations is shown to perform well in the problems considered. In the network design

problem of Section 2.10 we established that in systems comprising homogeneous station returns are NE-convex additional stations contribute diminishing returns under ESP.

Chapter 3

Dynamic Routing: Generalised "Join the Shortest Queue" Policies

3.1 Introduction

In the analysis of the static routing problem of the previous chapter, decisions regarding the destination of arriving jobs are based upon limited information. When the system controller only has knowledge of the arrival processes, service capabilities of the stations that comprise the network and the cost characteristics of the job classes we have seen that optimal static policies that arise in such situations take the form of simple Bernoulli routing policies. If the system controller knows the composition of the queueing jobs at each station at every routing decision epoch, then, combined with knowledge of the optimal static policy we would expect the controller to be able to make effective use of this information in the design of routing policies.

Here we consider how dynamic routing policies may be constructed for our complex multi-class routing problem which allows for many heterogeneous stations and local scheduling capability. We describe an approach to the development of dynamic routing heuristics by the enhancement of a static policy via the application of a single policy improvement step. For definiteness, we re-introduce the routing problem of Chapter 2 as a decision problem incorporating the dynamic structure of our heuristic routing policy. The problems and accompanying analysis are necessarily complex and so to keep the development as simple as possible we shall initially suppose that each service station may

be modelled as a multi-class $M/M/1$ queue. In fact, our approach is considerably more general than that, and we highlight an extension to the Klimov model in a later section. The major theoretical achievement of this chapter is the demonstration that the dynamic routing policies developed by this approach have a simple and intuitive structure which generalise "join the shortest queue" policies to our complex multi-class case in a natural way. In a computational investigation we compare the performance of the heuristic dynamic routing policies with a number of competitor policies. We conclude this chapter with an investigation into the application of these policies in a simple network design problem.

3.2 The Dynamic Routing Problem for Systems of Multi-Class $M/M/1$ Queues

We now return to the routing problem described in Chapter 2 but formulate it in a manner appropriate to the development of a dynamic routing heuristic. To reiterate, our problem concerns a distributed system that comprises a set of stations $\mathcal{M} = \{1, 2, \ldots, M\}$ and a system controller who routes the incoming jobs to stations on the basis of information received. Jobs from a number of different classes arrive at the system for processing. The job classes (and their constituent jobs) are either dedicated or generic. We assume that dedicated jobs arrive directly at their specified station for processing while the choice of station for generic jobs remains open. We denote the set of classes of generic jobs by G and the set of classes of jobs dedicated to station m by D_m, $m \in \mathcal{M}$. Hence $E = G \cup D_1 \cup \ldots \cup D_M$ is the set of job classes allowed access to the system while $E_m = G \cup D_m$ is the set of job classes allowed access to station $m \in \mathcal{M}$. Jobs arrive at the system in independent Poisson streams. Those dedicated to station m arrive there at rates $\{\lambda_{jm}\}_{j \in D_m}$. Generic jobs arrive at the system with rates $\{\lambda_g\}_{g \in G}$. Our routing problem concerns the choice of station to process the generic jobs. We formulate this as

a decision problem as follows:

(i) We write $\mathbf{N}(t) = \{\mathbf{N}_m(t)\}_{m\in\mathcal{M}}$ for the *state of the system* at time $t \in \mathbb{R}^+$, with $\mathbf{N}_m(t)$ for the *state of station m.* The latter is given by $\mathbf{N}_m(t) = \{N_{jm}(t)\}_{j\in G\cup D_m}$, where $N_{jm}(t)$ is the number of class j jobs present at station m (including any in service) at time t.

(ii) The *decision epochs* for the routing problem are the times at which generic jobs arrive at the system and will be the event times of a Poisson process with rate $\sum_{g\in G}\lambda_g$. We say that a *type g decision epoch* occurs when the generic arrival is from class $g \in G$. At a type g epoch, the actions available to the system controller are $A_g = \{a_1^g, a_2^g, \ldots, a_M^g\}$, where a_m^g denotes the routing of the newly arrived job to station m, thus increasing the class g queue length there by one.

(iii) The individual stations are modelled as (preemptive) multi-class $M/M/1$ queues. Between each decision epoch station m evolves via the arrival of dedicated jobs and service completions. Dedicated jobs from classes in D_m arrive at station m in independent Poisson stream with rates $\{\lambda_{jm}\}_{j\in D_m}$. A single server adopts an admissible scheduling policy $u_m \in \mathcal{U}_m$. The policy u_m is a simple priority policy in which jobs are served preemptively according to some fixed ordering among the job classes $G \cup D_m$. The service times for class j jobs are exponentially distributed with rate μ_{jm}, $j \in G \cup D_m$. We assume that all arrival and service processes are mutually independent.

(iv) A holding cost rate $c_j \geq 0$ is associated with each job class $j \in E$. Our goal is to develop *routing policies* (i.e. rules for making decisions regarding the routing of generic jobs) to minimise the long-run average holding cost rate

$$\sum_{m\in\mathcal{M}}\sum_{j\in G\cup D_m} c_j E(N_{jm}) \tag{3.1}$$

where the expectation in (3.1) is taken in steady state. We assume that the system is *stable*, in that a steady state solution with finite queue lengths exists.

In the following discussion we shall be interested in two classes of routing policies for the above problem. The class of static routing policies have been discussed in detail in Chapter 2. The optimisation problem there concerns the best choice of controller routing matrix $\mathbf{P} = (p_{gm})_{g\in G,\ m\in\mathcal{M}}$, with each station scheduling the arriving jobs according to the $c\mu$-rule. With any $\mathbf{P}$, all traffic arrives at each station m in independent Poisson streams. When this happens, the $c\mu$-rule minimises the holding cost rate at each station among all dynamic scheduling rules. In the present formulation of the routing problem, static policies will choose action a_m^g at a type g epoch with fixed probability p_{gm}, where $\sum_{m\in\mathcal{M}} p_{gm} = 1,\ g \in G$.

The class of dynamic policies are of primary interest here. Such policies can make use of the entire history of the system to date. However, the theory of stochastic dynamic programming asserts the existence of an optimal dynamic policy which is deterministic, stationary and Markov. Hence, an optimal decision at time $t \in \mathbb{R}^+$ is made with reference to $\mathbf{N}(t)$ only. The theoretical difficulties posed by our complex routing problem renders the chance of finding a simple closed form optimal solution to be very small. In addition, the combinatorial explosion means that a full DP solution would be difficult to obtain for any problems of reasonable size. The drawback of a purely numerical approach is that the solution is unlikely to provide any useful insight. This motivates a search for good heuristic policies.

The reader should note that in solving the dynamic routing problem we are in fact solving a joint routing/local scheduling problem which is fully dynamic. As noted above, optimal scheduling in a static routing regime is guaranteed by the $c\mu$-rule. While this is technically no longer the case when routing policies are dynamic, all our computations show that the degree of suboptimality in the local schedule is very small. That said, some applications may call for a choice of local scheduling policies u_m other than the $c\mu$-rule. For example, Ross and Yao (1991) give dedicated jobs some degree of priority over generic

jobs. This requirement can also be accommodated as a feature of the routing problem considered in the next chapter.

3.3 A Heuristic Dynamic Routing Policy for Routing to Multi-Class $M/M/1$ Queues

Here we develop dynamic routing heuristics by the application of a single dynamic programming policy improvement step to some given static policy $\mathbf{P}$. The natural choice of static policy is an optimal static policy, denoted $\mathbf{P}^*$, and we shall suppose that this is the case in what follows. We shall show that the dynamic routing policies which result from this approach in our complex multi-class environment are simple in structure and easily computable. They are natural developments of the "join the shortest queue" routing policy, extended in a way which is appropriate for a heterogeneous job population and a set of diverse stations. In pursuing this approach we develop an idea proposed in the context of simple single class systems by Krishnan (1987) and discussed by Tijms (1994). Their work considered the problem of routing a single class of job to a collection of $M/M/\eta$ queues with identical service rates at each station. We develop the dynamic routing heuristics as follows:

The policy improvement step utilises the quantities $\Delta^g(\mathbf{N}, m, \mathbf{P}^*)$ defined as follows:

In a situation in which the system state is $\mathbf{N}$ and a generic job of class g has just arrived, $\Delta^g(\mathbf{N}, m, \mathbf{P}^*)$ is the difference in total expected costs over an infinite horizon between the policy which allocates the class g job to station m and which then uses $\mathbf{P}^*$ and the static policy which uses $\mathbf{P}^*$ throughout.

In such a situation our heuristic policy π will route the newly arrived class g job to whichever station m has the smallest value of $\Delta^g(\mathbf{N}, m, \mathbf{P}^*)$. In the event of a tie, any of

the qualifying stations are chosen. We write

$$\pi(g, \mathbf{N}) = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \{\Delta^g(\mathbf{N}, m, \mathbf{P}^*)\} \tag{3.2}$$

where $\pi(g, \mathbf{N})$ is used for the station chosen by policy π for the class g job when the system state is $\mathbf{N}$.

Since under the static policy $\mathbf{P}^*$ the processes at the M stations evolve independently, the policy π may be described more simply in terms of station-specific quantities $\Delta_m^g(\mathbf{N}_m)$ defined as follows:

Consider station m in isolation, evolving under its local scheduling policy u_m with generic arrival rates $\{p_{gm}^* \lambda_g\}_{g \in G}$ determined by $\mathbf{P}^*$. The quantity $\Delta_m^g(\mathbf{N}_m)$ is defined to be the difference in total expected costs over an infinite horizon for station m alone between starting in state $\mathbf{N}_m + 1_m^g$ and in state $\mathbf{N}_m$. We use 1_m^g to denote a $|G \cup D_m|$-vector whose g^{th} component is 1 with zeroes elsewhere. By a simple conditioning argument we deduce that

$$\Delta^g(\mathbf{N}, m, \mathbf{P}^*) = -\sum_{m' \in \mathcal{M}} p_{gm'}^* \Delta_{m'}^g(\mathbf{N}_{m'}) + \Delta_m^g(\mathbf{N}_m)$$

and so (3.2) may be re-expressed as

$$\pi(g, \mathbf{N}) = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \{\Delta_m^g(\mathbf{N}_m)\}. \tag{3.3}$$

It should be clear from (3.3) that policy π will never route a class j job to station m when $\mu_{jm} = 0$ since any corresponding Δ_m^j is infinite. Hence, without loss of generality we can assume that $\mu_{jm} > 0$ for all $j,\ m$ in what follows.

Under the routing heuristic given by (3.3), each station has an index dependent both upon its current state and the class of job to be allocated. The station chosen by the policy is the one with the smallest index value. The advantages of (3.3) over (3.2) for computation are clear from the reduction in dimensionality of the state variables concerned.

The key quantities $\Delta_m^g(\mathbf{N}_m)$ are readily available and are expressed in terms of the so-called *relative costs* (or sometimes *biases*) associated with station m (evolving under scheduling rule u_m and with generic arrival rates $p_{jm}^* \lambda_j^G$), regarded as an undiscounted Markov decision process (MDP). The relative costs are state-valued quantities which reflect the transient effect of the starting state on the total expected costs for station m under local scheduling rule u_m and static policy $\mathbf{P}^*$. Use $\mathbf{0}_m$ to denote the zero-state when station m is empty and $\Psi_m(\mathbf{P}^*)$ to denote the long-run average cost per unit time for station m under static policy $\mathbf{P}^*$. From standard MDP theory (see, for example, Tijms (1994)) we define the relative costs for station m, denoted $h_m(\mathbf{N}_m)$, by

$$h_m(\mathbf{N}_m) = K_m(\mathbf{N}_m) - \Psi_m(\mathbf{P}^*) T_m(\mathbf{N}_m), \tag{3.4}$$

where $K_m(\mathbf{N}_m)$ is defined as the total expected cost incurred by station m from initial state $\mathbf{N}_m$ until it first enters state $\mathbf{0}_m$ (i.e., until it first becomes empty) and $T_m(\mathbf{N}_m)$ is the corresponding expected time. Tijms (1994) provides an economic interpretation of the relative costs. Consider any two states $\mathbf{N}_m$ and $\mathbf{N}'_m$, $h_m(\mathbf{N}_m) - h_m(\mathbf{N}'_m)$ is the difference in total expected costs over an infinite horizon at station m by starting in state $\mathbf{N}_m$ rather than in state $\mathbf{N}'_m$ under local scheduling rule u_m and static policy $\mathbf{P}^*$. From this interpretation of the relative costs and the definition of the $\Delta_m^g(\mathbf{N}_m)$ we have that

$$\begin{aligned}
\Delta_m^g(\mathbf{N}_m) &= h_m(\mathbf{N}_m + \mathbf{1}_m^g) - h_m(\mathbf{N}_m) \\
&= \{K_m(\mathbf{N}_m + \mathbf{1}_m^g) - K_m(\mathbf{N}_m)\} \\
&\quad - \Psi_m(\mathbf{P}^*)\{T_m(\mathbf{N}_m + \mathbf{1}_m^g) - T_m(\mathbf{N}_m)\}.
\end{aligned} \tag{3.5}$$

The quantities $K_m(\mathbf{N}_m)$ and $T_m(\mathbf{N}_m)$ may be calculated via a simple one-step value iteration argument, here it is beneficial to consider both arrival and service completion epochs. Suppose that local scheduling rule u_m at station m processes a job from class

$j(\mathbf{N}_m) \in G \cup D_m$ in state $\mathbf{N}_m$. We then have

$$\left\{\sum_{g\in G} p^*_{gm}\lambda_g + \sum_{j\in D_m}\lambda_{jm} + \mu_{j(\mathbf{N}_m)m}\right\} K_m(\mathbf{N}_m)$$

$$= \sum_{j\in G\cup D_m} c_j N_{jm} + \sum_{g\in G} p^*_{gm}\lambda_g K_m\left(\mathbf{N}_m + 1^g_m\right) + \sum_{j\in D_m}\lambda_{jm}K_m\left(\mathbf{N}_m + 1^j_m\right) \qquad (3.6)$$

$$+ \mu_{j(\mathbf{N}_m)m}K_m(\mathbf{N}_m - 1^{j(\mathbf{N}_m)}_m)$$

and

$$\left\{\sum_{g\in G} p^*_{gm}\lambda_g + \sum_{j\in D_m}\lambda_{jm} + \mu_{j(\mathbf{N}_m)m}\right\} T_m(\mathbf{N}_m)$$

$$= 1 + \sum_{g\in G} p^*_{gm}\lambda_g T_m\left(\mathbf{N}_m + 1^g_m\right) + \sum_{j\in D_m}\lambda_{jm}T_m\left(\mathbf{N}_m + 1^j_m\right) \qquad (3.7)$$

$$+ \mu_{j(\mathbf{N}_m)m}T_m(\mathbf{N}_m - 1^{j(\mathbf{N}_m)}_m).$$

The solution of these recursions yields the key index values $\Delta^g_m(\mathbf{N}_m)$ via (3.5).

We now assert that the indices $\Delta^g_m(\mathbf{N}_m)$ are simple congestion measures which are linear in the queue lengths $\mathbf{N}_m$. The following discussion will be simplified if in what follows we extend the definition of $\Delta^g_m(\mathbf{N}_m)$ to all $j \in G \cup D_m$ via the expression in (3.5).

Theorem 24 (Linear Characterisation of $\Delta^j_m(\mathbf{N}_m)$)

For each station $m \in \mathcal{M}$ there exists a matrix $\Theta^m = \{\theta^m_{jk}\}_{j,k\in G\cup D_m}$ and a vector $\Omega^m = \{\omega^m_j\}_{j\in G\cup D_m}$ such that

$$\Delta^j_m(\mathbf{N}_m) = \sum_{k\in G\cup D_m}\theta^m_{jk}N_{km} + \omega^m_j,\ j \in G\cup D_m,\ m\in\mathcal{M}.$$

Proof

To minimise notational complexities, we fix $m \in \mathcal{M}$ and where possible drop m from the notation. We shall suppose that job classes at station m are numbered according to the priority accorded them by local scheduling rule u_m, i.e. that u_m imposes the class

priorities $1 \rightarrow 2 \rightarrow \ldots \rightarrow |G \cup D_m|$. $K(\mathbf{N})$ now stands for the total expected cost incurred in emptying station m from an initial state $\mathbf{N}$ when the station operates under u_m while receiving generic traffic at rates $\{p^*_{gm}\lambda_g\}_{g \in G}$. The key to the proof is in demonstrating that $K(\mathbf{N})$ is quadratic in $\mathbf{N}$ in the sense that

$$\mathbf{N} = \sum_{j=1}^{|G \cup D_m|} N_j 1^j \Rightarrow K(\mathbf{N}) = \sum_{k \in G \cup D_m} \sum_{l \in G \cup D_m} K_{kl} N_k N_l + \sum_{k \in G \cup D_m} K_k N_k \qquad (3.8)$$

for some constants $\{K_{kl}\}_{k \in G \cup D_m, l \in G \cup D_m}$ and $\{K_k\}_{k \in G \cup D_m}$. As before, we use 1^j to denote a $|G \cup D_m|$-vector whose j^{th} component is 1, with zeroes elsewhere.

We shall prove (3.8) by an induction on Γ, the job class of highest priority under u_m represented in $\mathbf{N}$. Firstly, consider the case $\Gamma = |G \cup D_m|$, i.e., $\mathbf{N} = N1^{|G \cup D_m|}$ for some positive integer N. As a preliminary, focus on the initial state $1^{|G \cup D_m|}$. We write $\tilde{T}_{|G \cup D_m|}$ for the random time it takes to empty the system from $1^{|G \cup D_m|}$ and $\tilde{K}(1^{|G \cup D_m|})$ for the corresponding random cost. It must then follow that, if we now begin in state $N1^{|G \cup D_m|}$ and process jobs until the station enters state $(N-1)1^{|G \cup D_m|}$ for the first time, then the time elapsed is $\tilde{T}_{N|G \cup D_m|}$ and the cost incurred is

$$(N-1)c_{|G \cup D_m|}\tilde{T}_{N|G \cup D_m|} + \tilde{K}_N(1^{|G \cup D_m|}), \qquad (3.9)$$

where

$$\tilde{T}_{N|G \cup D_m|} \underset{dn}{=} \tilde{T}_{|G \cup D_m|} \text{ and } \tilde{K}_N(1^{|G \cup D_m|}) \underset{dn}{=} \tilde{K}(1^{|G \cup D_m|}). \qquad (3.10)$$

The notation "$\underset{dn}{=}$" in (3.10) indicates that the random variables concerned have the same probability distribution. To understand (3.9) and (3.10) imagine that $(N-1)$ of the $|G \cup D_m|$-jobs present at time 0 are laid on one side and play no part in the processing. These jobs will incur a cost $(N-1)c_{|G \cup D_m|}\tilde{T}_{N|G \cup D_m|}$. Otherwise, the situation is stochastically identical to that beginning from $1^{|G \cup D_m|}$ above. We now empty the system by considering successive first time transitions $N1^{|G \cup D_m|} \rightarrow (N-1)1^{|G \cup D_m|} \rightarrow (N-2)1^{|G \cup D_m|} \rightarrow \ldots \rightarrow$

$1^{|G\cup D_m|} \rightarrow \mathbf{0}$. By repetition of the argument to (3.9) and (3.10) we can write the total cost incurred in emptying the system from $N1^{|G\cup D_m|}$ as

$$\sum_{j=1}^{N}(j-1)c_{|G\cup D_m|}\tilde{T}_{j|G\cup D_m|} + \sum_{j=1}^{N}\tilde{K}_j(1^{|G\cup D_m|}). \tag{3.11}$$

In (3.11), the random variables $\tilde{T}_{j|G\cup D_m|}$, $1 \leq j \leq N$, are i.i.d. as are $\tilde{K}_j(1^{|G\cup D_m|})$, $1 \leq j \leq N$. The corresponding expectations are $T(1^{|G\cup D_m|})$ and $K(1^{|G\cup D_m|})$ respectively. From (3.11) we deduce that the expected cost incurred in emptying the system from $N1^{|G\cup D_m|}$ is

$$\begin{aligned} K(N1^{|G\cup D_m|}) &= E\left\{\sum_{j=1}^{N}(j-1)c_{|G\cup D_m|}\tilde{T}_{j|G\cup D_m|} + \sum_{j=1}^{N}\tilde{K}_j(1^{|G\cup D_m|})\right\} \\ &= \frac{1}{2}(N-1)Nc_{|G\cup D_m|}T(1^{|G\cup D_m|}) + NK(1^{|G\cup D_m|}), \end{aligned}$$

which is quadratic in N. The inductive hypothesis is established for the case $\Gamma = |G\cup D_m|$.

We now suppose that the inductive hypothesis holds whenever initial state N is such that $k+1 \leq \Gamma \leq |G \cup D_m|$ and consider the case $\Gamma = k$. Hence the initial state N is assumed to be

$$\mathrm{N} = N1^k + \sum_{j=k+1}^{|G\cup D_m|} N_j 1^j \tag{3.12}$$

with $N > 0$. As a preliminary, focus on an initial state 1^k. We write $\tilde{T}_k$ for the random time it takes to empty the system of all jobs from class k (and hence also from classes $1, 2, \ldots, k-1$) for the first time and $\tilde{K}(1^k)$ for the corresponding random cost. At time $\tilde{T}_k$ the (random) state of the system is written

$$\sum_{j=k+1}^{|G\cup D_m|} \tilde{N}_j 1^j \tag{3.13}$$

and comprises jobs from classes $k+1, k+2, \ldots, |G \cup D_m|$ which arrived at the system

during $[0, \tilde{T}_k)$. It must then follow that, if we now begin in state $\mathbf{N}$ in (3.12) and process jobs until the number of class k jobs drops to $(N-1)$ for the first time, then the time elapsed is $\tilde{T}_{Nk}$ and the cost incurred is

$$\left\{(N-1)c_k + \sum_{j=k+1}^{|G\cup D_m|} N_j c_j\right\} \tilde{T}_{Nk} + \tilde{K}_N(1^k), \tag{3.14}$$

where

$$\tilde{T}_{Nk} \underset{dn}{=} \tilde{T}_k \text{ and } \tilde{K}_N(1^k) \underset{dn}{=} \tilde{K}(1^k). \tag{3.15}$$

The reasoning which yields (3.14) and (3.15) is a simple development of the argument following (3.10). Additionally, the (random) state of the system at time $\tilde{T}_{Nk}$ may be written

$$(N-1)1^k + \sum_{j=k+1}^{|G\cup D_m|} (N_j + \tilde{N}_{Nj})1^j \tag{3.16}$$

where

$$\left(\tilde{N}_{N(k+1)}, \tilde{N}_{N(k+2)}, \ldots, \tilde{N}_{N|G\cup D_m|}\right) \underset{dn}{=} \left(\tilde{N}_{(k+1)}, \tilde{N}_{(k+2)}, \ldots, \tilde{N}_{|G\cup D_m|}\right).$$

We now empty the system of class k jobs by considering successive transitions

$$\begin{aligned} N1^k + \sum_{j=k+1}^{|G\cup D_m|} N_j 1^j \;&\rightarrow\; (N-1)1^k + \sum_{j=k+1}^{|G\cup D_m|} (N_j + \tilde{N}_{Nj})1^j \rightarrow \ldots\ldots \\ \ldots\ldots \;&\rightarrow\; \sum_{j=k+1}^{|G\cup D_m|} (N_j + \sum_{l=1}^{N} \tilde{N}_{lj})1^j. \end{aligned} \tag{3.17}$$

By suitable repetition of the argument to (3.14) and (3.15) we can write the total cost

incurred in emptying the system of class k jobs from state $\mathbf{N}$ in (3.12) as

$$\sum_{l=1}^{N}\left\{(l-1)c_k+\sum_{j=k+1}^{|G\cup D_m|}(N_j+\sum_{n=l+1}^{N}\tilde{N}_{nj})c_j\right\}\tilde{T}_{lk}+\sum_{l=1}^{N}\tilde{K}_l(1^k). \tag{3.18}$$

In (3.17) and (3.18), the $(\tilde{N}_{l(k+1)},\tilde{N}_{l(k+2)},\ldots,\tilde{N}_{l|G\cup D_m|})$, $1\leq l\leq N$, are i.i.d. as are the $\tilde{T}_{lk}$, $1\leq l\leq N$, and the $\tilde{K}_l(1^k)$, $1\leq l\leq N$. The respective means are written $(\hat{N}_{(k+1)},\hat{N}_{(k+2)},\ldots,\hat{N}_{|G\cup D_m|})$, $\hat{T}_k$ and $\hat{K}(1^k)$. We also note that $\sum_{l'=l+1}^{N}\tilde{N}_{l'j}$ and $\tilde{T}_{lk}$ are independent for all choices of j,l. From (3.17) and (3.18) we deduce that the expected cost incurred in emptying the system from $\mathbf{N}$ is

$$\begin{aligned}K(\mathbf{N})&=K\left(N1^k+\sum_{j=k+1}^{|G\cup D_m|}N_j1^j\right)\\&=\frac{1}{2}(N-1)Nc_k\hat{T}_k+\sum_{j=k+1}^{|G\cup D_m|}N\left\{N_j+\frac{1}{2}(N-1)\hat{N}_j\right\}c_j\hat{T}_k+N\hat{K}(1^k)\\&\quad+E\left[K\left\{\sum_{j=k+1}^{|G\cup D_m|}(N_j+\sum_{l=1}^{N}\tilde{N}_{lj})1^j\right\}\right].\end{aligned} \tag{3.19}$$

Consider the system state which is the argument of K in the last term on the r.h.s. in (3.19). The job class of highest index represented is $k+1$ and the inductive hypothesis applies. It follows straightforwardly that the final term on the r.h.s. of (3.19) is quadratic in $\mathbf{N}$ as, plainly, are the first three terms. Hence $K(\mathbf{N})$ is quadratic in $\mathbf{N}$ for all choices of $\mathbf{N}$.

Now let $T(\mathbf{N})$ be the total expected time taken emptying station m from initial state $\mathbf{N}$ when the station operates under u_m while receiving generic traffic according to static policy $\mathbf{P}^*$. A similar (though less involved) induction argument yields the conclusion that $T(\mathbf{N})$ is linear in $\mathbf{N}$ for all choices of $\mathbf{N}$. We may thus write that

$$\mathbf{N}=\sum_{j=1}^{|G\cup D_m|}N_j1^j\Rightarrow T(\mathbf{N})=\sum_{j\in G\cup D_m}t_jN_j \tag{3.20}$$

for some constants $\{t_j\}_{j\in G\cup D_m}$. To see this, if we follow the induction argument above for the case $\Gamma = |G \cup D_m|$, we deduce that the expected time taken to empty the system from $N1^{|G\cup D_m|}$ is

$$T(N1^{|G\cup D_m|}) = E\left\{\sum_{j=1}^{N} \tilde{T}_{j|G\cup D_m|}\right\} = NT(1^{|G\cup D_m|}),$$

which is linear in N. This establishes the inductive hypothesis for the case $\Gamma = |G \cup D_m|$. Continuing, we easily conclude from the analysis that for case $\Gamma = k$, where the initial state $\mathbf{N}$ is given by (3.12), the expected time taken emptying the system from state $\mathbf{N}$ is

$$\begin{aligned} T(\mathbf{N}) &= T\left(N1^k + \sum_{j=k+1}^{|G\cup D_m|} N_j 1^j\right) \\ &= N\hat{T}_k + E\left[T\left\{\sum_{j=k+1}^{|G\cup D_m|}(N_j + \sum_{l=1}^{N}\tilde{N}_{lj})1^j\right\}\right]. \end{aligned} \tag{3.21}$$

In the last term on the r.h.s. of (3.21) the job of highest index represented is $k+1$ and the inductive hypothesis applies. Hence, $T(\mathbf{N})$ is linear in $\mathbf{N}$ for all choices of $\mathbf{N}$. Note that the absence of constant terms in (3.8) and (3.20) is accounted for by the fact that $K(\mathbf{0}) = T(\mathbf{0}) = 0$.

We now restore the station suffix m fully to the notation. We have established that $K_m(\mathbf{N}_m)$ is quadratic and $T_m(\mathbf{N}_m)$ is linear in $\mathbf{N}_m$. It now follows immediately from the expression on the r.h.s. of (3.5) that, for all $j \in G \cup D_m$, $\Delta_m^j(\mathbf{N}_m)$ must be linear in $\mathbf{N}_m$.

□

In the light of Theorem 24, we may now re-express (3.3), which characterises our routing heuristic π, as

$$\pi(g, \mathbf{N}) = \underset{m\in\mathcal{M}}{\operatorname{argmin}}\left\{\sum_{k\in G\cup D_m} \theta_{gk}^m N_{km} + \omega_g^m\right\}. \tag{3.22}$$

A computational advantage of the routing heuristic's linear index structure is that only

sufficient evaluations of the $\Delta_m^j(\mathbf{N}_m)$ are required to determine the coefficients $\{\theta_{jk}^m\}_{j,k\in G\cup D_m}$ and $\{\omega_j^m\}_{j\in G\cup D_m}$. In our work such evaluations utilised (3.5) together with a value iteration approach to the computation of the required $K_m(\mathbf{N}_m)$ and $T_m(\mathbf{N}_m)$ via (3.6) and (3.7) respectively. In most problems this represents a huge computational saving over full DP.

Theorem 25 contains further information on the matrix Θ^m. Firstly it is symmetric and hence has only $\frac{1}{2}|G\cup D_m|(|G\cup D_m|+1)$ distinct entries. Second, all its entries are non-negative, implying that each $\Delta_m^g(\mathbf{N}_m)$ on the r.h.s. of (3.22) is indeed a congestion measure in the sense of being increasing in the queue lengths $\mathbf{N}_m$.

Theorem 25

The matrices Θ^m, $m\in\mathcal{M}$, which determine the indices $\Delta_m^j(\mathbf{N}_m)$ are such that

(a) $\theta_{jk}^m=\theta_{kj}^m$ *for all* $j,k\in G\cup D_m$, $m\in\mathcal{M}$; *(symmetry)*

(b) $\theta_{jk}^m\geq 0$ *for all* $j,k\in G\cup D_m$, $m\in\mathcal{M}$. *(non-negativity)*

Proof

From the expression for $\Delta_m^j(\mathbf{N}_m)$ in the statement of Theorem 24 it is straightforward to conclude that, for all $j,k\in G\cup D_m$ and all $m\in\mathcal{M}$,

$$\begin{aligned}\theta_{jk}^m &= \Delta_m^j(1_m^k)-\Delta_m^j(0_m)\\ &= \left\{K_m(1_m^k+1_m^j)-K_m(1_m^k)-K_m(1_m^j)\right\}\\ &\qquad -\Psi_m(\mathbf{P}^*)\left\{T_m(1_m^k+1_m^j)-T_m(1_m^k)-T_m(1_m^j)\right\} \quad (3.23)\\ &= K_m(1_m^k+1_m^j)-K_m(1_m^k)-K_m(1_m^j) \quad (3.24)\\ &= \theta_{kj}^m,\end{aligned}$$

by the symmetry between j and k in (3.24). In order to obtain (3.23) from (3.5) we have used the fact that $K_m(0_m)=T_m(0_m)=0$. To obtain (3.24), we have used the linearity of

$T_m(\mathbf{N}_m)$ established in the proof of Theorem 24 and expressed in (3.20) to conclude that

$$T_m(\mathbf{1}_m^k + \mathbf{1}_m^j) - T_m(\mathbf{1}_m^k) - T_m(\mathbf{1}_m^j) = 0.$$

This establishes (a).

From (3.24), it is enough to prove (b) that, for all choices of j, k and m,

$$K_m(\mathbf{1}_m^k + \mathbf{1}_m^j) \geq K_m(\mathbf{1}_m^k) + K_m(\mathbf{1}_m^j). \tag{3.25}$$

Consider station m in initial state $\mathbf{1}_m^k + \mathbf{1}_m^j$, processed according to scheduling rule u_m until it reaches the empty state $\mathbf{0}_m$. In an obvious way, the system evolution may be regarded as the (disjoint union of) two branching processes emanating respectively from the class k and the class j jobs present initially. Suppose that holding costs are reduced such that when processing is allocated to the k-branching process, *only* costs from jobs within that process are incurred, and similarly for the j-branching process. Under this cost reduction, the total expected cost incurred until the system empties is evidently $K_m(\mathbf{1}_m^k) + K_m(\mathbf{1}_m^j)$, and the inequality (3.25) follows. □

We obtain further simplification of the indices $\Delta_m^j(\mathbf{N}_m)$ in the important special case where the traffic at each station is *stochastically indistinguishable*. Recall from Section 2.8.1 we say that the jobs (both generic and dedicated) at station m are stochastically indistinguishable if $\mu_{jm} = \mu_m,\ j \in G \cup D_m$. Theorem 26 asserts that in this case, the symmetric matrix Θ^m has a special structure and just $|G \cup D_m|$ distinct elements. Hence significant computational savings are available in this case. It will simplify the discussion if we fix $m \in \mathcal{M}$ and suppose that at station m, local scheduling rule u_m imposes the priorities $1 \to 2 \to \ldots \to |G \cup D_m|$. We also write λ_{jm} for the arrival rate at station m for all job classes j, i.e. $\lambda_{gm} = p_{gm}^* \lambda_g,\ g \in G$.

Theorem 26 (The Stochastically Indistinguishable Case)

If the jobs at station m are stochastically indistinguishable then the matrix Θ^m is such

that

$$\theta_{jk}^m = \theta_{kj}^m = \theta_j^m,\ 1 \leq k \leq j \leq |G \cup D_m|, \tag{3.26}$$

and the $|G \cup D_m|$ *constants* $\theta_j^m,\ 1 \leq j \leq |G \cup D_m|$, *are determined by the recursive scheme*

$$\theta_j^m = \left(c_j + \sum_{l=j+1}^{|G\cup D_m|} \lambda_{lm}\theta_l^m\right)\left(\mu_m - \sum_{l=1}^{j}\lambda_{lm}\right)^{-1},\ 1 \leq j \leq |G \cup D_m| - 1, \tag{3.27}$$

$$\theta_{|G\cup D_m|}^m = c_{|G\cup D_m|}T_m(1_m^{|G\cup D_m|}). \tag{3.28}$$

Proof

Fix j and k such that $1 \leq k \leq j \leq |G \cup D_m|$. We use (3.6) and (3.7) to develop expressions for $K_m(1_m^k + 1_m^j)$, $K_m(1_m^k)$, $T_m(1_m^k + 1_m^j)$ and $T_m(1_m^k)$. From these we can utilise (3.5) to infer the following expression for $\Delta_m^j(1_m^k)$:

$$\left(\sum_{l=1}^{|G\cup D_m|}\lambda_{lm} + \mu_m\right)\Delta_m^j(1_m^k) = c_j + \sum_{l=1}^{|G\cup D_m|}\lambda_{lm}\Delta_m^j(1_m^k + 1_m^l) + \mu_m\Delta_m^j(0_m). \tag{3.29}$$

From (3.29) we obtain

$$\mu_m\theta_{jk}^m = c_j + \sum_{l=1}^{|G\cup D_m|}\lambda_{lm}\theta_{jl}^m, \tag{3.30}$$

which establishes (3.26). Now allow j to take any value in the range $1 \leq j \leq |G \cup D_m|$. By appeal to the symmetry of Θ^m, established in Theorem 25(a) and from (3.26) we deduce that for any such j,

$$\theta_{jl}^m = \begin{cases} \theta_j^m, & 1 \leq l \leq j, \\ \theta_l^m, & j+1 \leq l \leq |G \cup D_m|. \end{cases} \tag{3.31}$$

We now use (3.31) within (3.30) to obtain (3.27). Finally, we use (3.24) to infer that

$$\theta^m_{|G\cup D_m|} = K_m(21^{|G\cup D_m|}_m) - 2K_m(1^{|G\cup D_m|}_m)$$

and (3.28) follows easily from an analysis of the costs incurred in emptying station m from initial state $21^{|G\cup D_m|}_m$. □

A particularly simple set up for the stochastically indistinguishable case is when the system comprises homogeneous stations. Suppose that a generic job belonging to the class of lowest priority under the local scheduling rule arrives at the system. According to Theorem 26 the form of the station m index in this case is

$$\theta(\text{number of jobs at station } m) + \omega.$$

Hence in this scenario, our routing heuristic will send the incoming job to the station with the smallest (total) number of jobs present, i.e. it will join the shortest queue.

3.3.1 Extension to the Klimov Network Model

In the development of the dynamic heuristic routing policy we have only considered modelling the stations comprising the network as multi-class $M/M/1$ queues, operating under a simple priority policy for local scheduling. We are not restricted in our usage to this particular situation and a range of extensions to more general scenarios are readily available. Here, we consider the case in which each station m is modelled as a Klimov network. (See Section 2.2, Example 1 for details of this model). The analysis above continues to hold when the individual stations are modelled as a Klimov network . However, we shall now highlight the main points from these analyses.

Slight adjustments to proof of Theorem 24 incorporating the feedback mechanism of

the Klimov model again yields

$$\Delta_m^j(\mathbf{N}_m) = \sum_{k \in G \cup D_m} \theta_{jk}^m N_{km} + \omega_j^m, \ j \in G \cup D_m, \ m \in \mathcal{M}. \tag{3.32}$$

Hence, when the individual stations are modelled as a Klimov network the key indices $\Delta_m^j(\mathbf{N}_m)$ are also linear in the state of station m. The coefficients $\{\theta_{jk}^m\}_{j,k \in G \cup D_m}$ and $\{\omega_j^m\}_{j \in G \cup D_m}$ in (3.32) may be computed via forms of the recursions (3.6) and (3.7) appropriate to the Klimov network case and are given by

$$\begin{aligned}
&\left\{ \sum_{g \in G} p_{gm}^* \lambda_g + \sum_{j \in D_m} \lambda_{jm} + \mu_{j(\mathbf{N}_m)m} \right\} K_m(\mathbf{N}_m) \\
&= \sum_{j \in G \cup D_m} c_j N_{jm} + \sum_{g \in G} p_{gm}^* \lambda_g K_m\left(\mathbf{N}_m + 1_m^g\right) + \sum_{j \in D_m} \lambda_{jm} K_m\left(\mathbf{N}_m + 1_m^j\right) \\
&\quad + \mu_{j(\mathbf{N}_m)m} \left\{ q_{j(\mathbf{N}_m)0}^m K_m(\mathbf{N}_m - 1_m^{j(\mathbf{N}_m)}) + \sum_{k \in G \cup D_m} q_{j(\mathbf{N}_m)k}^m K_m(\mathbf{N}_m - 1_m^{j(\mathbf{N}_m)} + 1_m^k) \right\}
\end{aligned}$$

and

$$\begin{aligned}
&\left\{ \sum_{g \in G} p_{gm}^* \lambda_g + \sum_{j \in D_m} \lambda_{jm} + \mu_{j(\mathbf{N}_m)m} \right\} T_m(\mathbf{N}_m) \\
&= 1 + \sum_{g \in G} p_{gm}^* \lambda_g T_m\left(\mathbf{N}_m + 1_m^g\right) + \sum_{j \in D_m} \lambda_{jm} T_m\left(\mathbf{N}_m + 1_m^j\right) \\
&\quad + \mu_{j(\mathbf{N}_m)m} \left\{ q_{j(\mathbf{N}_m)0}^m T_m(\mathbf{N}_m - 1_m^{j(\mathbf{N}_m)}) + \sum_{k \in G \cup D_m} q_{j(\mathbf{N}_m)k}^m T_m(\mathbf{N}_m - 1_m^{j(\mathbf{N}_m)} + 1_m^k) \right\}.
\end{aligned}$$

The solution of these recursions yields the key indices $\Delta_m^j(\mathbf{N}_m)$ via (3.5).

Theorem 25 still holds in this case, thus the matrix of coefficients Θ^m is symmetric and has non-negative entries. Further simplification of the key indices is available for the Klimov model when all job classes are stochastically indistinguishable at each station (i.e., $\mu_{jm} = \mu_m, \ j \in G \cup D_m$, and $q_{ik}^m = q_{jk}^m = q_k^m$ for all choices of i, j and $k \in G \cup D_m$). An equivalent version of Theorem 26 is available where the $|G \cup D_m|$ constants θ_j^m, $1 \leq$

$j \leq |G \cup D_m|$ are now determined by the recursive scheme

$$\theta_j^m = \left(c_j + \sum_{l=j+1}^{|G\cup D_m|} \lambda_{lm}\, \theta_l^m + \mu_m \sum_{l=j+1}^{|G\cup D_m} q_l^m \theta_l^m \right)$$
$$\times \left(\mu_m (1 - \sum_{l=1}^{j} q_l^m) - \sum_{l=1}^{j} \lambda_{lm} \right)^{-1}, \; 1 \leq j \leq |G \cup D_m| - 1,$$

$$\theta_{|G\cup D_m|}^m = c_{|G\cup D_m|} T_m(1_m^{|G\cup D_m|}).$$

Straightforward amendments to the proof of Theorem 26 yield the desired result. The comments following Theorem 26 regarding a set of homogeneous station with stochastically indistinguishable job classes at each station apply equally to the Klimov model. Hence, an arriving generic job of lowest priority under the local scheduling rule u_m will be routed to the shortest queue.

3.4 Numerical Study

We now present the results of a computational study of the comparative performance of our routing heuristic with a number of competitor policies. Technical considerations restrict the scope of our investigation to problems where the calculation of a fully optimal dynamic routing policy via DP is computationally feasible. We focus on two simple scenarios involving two homogeneous stations each admitting two generic job classes. These scenarios are sufficiently simple to allow for the DP calculations yet are sufficiently rich to provide insight. Theorem 26 (and the following comment) allude to simplifications of the key indices Δ_m^g when job classes are stochastically indistinguishable. We shall initially concern ourselves with this special case. The second part of the study extends from this simple case and allows for increasing dissimilarities in the service and cost characteristics of the arriving job classes. The following routing policies (all identified by

single letter abbreviations) are considered for our calculations:

Optimal Static Policy (S): Incoming jobs are routed to the two stations according some fixed probability provided by $\mathbf{P}^*$, the optimal controller routing matrix;

Equal Splitting Policy (E): Incoming jobs are routed to the two stations with equal probability. Under E, each station receives traffic at rate $(\frac{\lambda_1}{2}, \frac{\lambda_2}{2})$;

Round Robin (R): Incoming jobs are sent alternately to the two stations. Each station now faces an arrival stream which is a renewal process with i.i.d. gamma $\Gamma(2, \lambda_1+\lambda_2)$ interarrival times. The only information requirement for the policy is the destination of the last arrival;

Join the Shortest Queue (J): Incoming jobs are sent to whichever station has the smaller total number of jobs present and to either station in the event of a tie;

Routing Heuristic (H): This is the heuristic developed in the previous section by implementing a policy improvement step to the optimal static policy. The information requirement is marginally greater than for J in that the queue lengths for each job class at each station are needed;

Optimal Dynamic Routing Policy (O): This is developed via DP. Although it has the same information requirements as H it will in general be much more difficult to compute and considerably more complex in structure.

3.4.1 Stochastically Indistinguishable Job Classes

We shall suppose that both stations are preemptive two-class $M/M/1$ systems operating the $c\mu$-rule for local scheduling. When job classes are stochastically indistinguishable the competing jobs have identical processing requirements and we take $\mu_1 = \mu_2 = 1$ for the completion rates of the exponential service rates of the two job classes. Hence the $c\mu$-rule accords priority to the job class with the larger holding cost rate. We fix $c_2 = 1$ and

take $c_1 \geq 1$, ensuring that class 1 always has higher priority. Theorem 26 applies and the matrix of coefficients determining the indices Δ_m^g will have the form

$$\Theta^1 = \Theta^2 = \begin{pmatrix} \theta_1 & \theta_2 \\ \theta_2 & \theta_2 \end{pmatrix}.$$

Further, it is straightforward to show that $\theta_1 \to \theta_2$ as $c_1 \searrow c_2 = 1$ and hence our routing heuristic becomes "join the shortest queue" in the single job class limit. We shall consider the E, R, J, H heuristics and policy O for such problems. Recall that from Corollary 17 when job classes are stochastically indistinguishable the optimal station return Ψ_m is convex. By Theorem 20 when stations are homogeneous and the station returns are convex, E coincides with the optimal static policy S.

The long-run average holding cost rates incurred under these policies were computed for problems with c_1 ranging from 1.5 to 10 and $\rho = \lambda_1 + \lambda_2$ taking the values 0.6 (light traffic), 1.0 (moderate traffic) and 1.5 (moderate to heavy traffic). For each (c_1, ρ)-combination, 100 (λ_1, λ_2) pairs were generated by sampling λ_1 from a $U[0.2, \rho - 0.2]$ distribution then choosing $\lambda_2 = \rho - \lambda_1$. For each generated problem the cost rates C^E (the long-run average cost rate under E), C^R, C^J, C^H and C^O were computed. A simple closed form expression for C^E is available. For the other policies an appropriate form of DP value iteration was used to compute costs. Theoretical considerations indicate that $C^E \geq C^H \geq C^O$. In fact, in all 1200 problems studied, it emerged that

$$C^E \geq C^R \geq C^J \geq C^H \geq C^O \tag{3.33}$$

and so costs decrease as the information requirements of the policies grow. In light of (3.33), we present our results in terms of the percentage cost degradation experienced as we move from right to left through the inequalities in (3.33). For example, we write

$$\Delta(E, R) = 100\{(C^E - C^R)/C^R\}$$

and similarly for $\Delta(R, J)$, $\Delta(J, H)$ and $\Delta(H, O)$. In almost all cases, the samples of size 100 produced little within-sample variation in these quantities and so in Table 3.1 we simply report median values.

(c_1, ρ)	$\Delta(E, R)$	$\Delta(R, J)$	$\Delta(J, H)$	$\Delta(H, O)$
1.5, 0.6	15.681	5.712	0.346	0.056
3.0, 0.6	12.810	4.684	1.021	0.025
5.0, 0.6	11.122	4.101	1.410	0.005
10.0, 0.6	9.555	3.557	1.758	0.003
1.5, 1.0	21.622	12.215	0.419	0.254
3.0, 1.0	18.115	10.084	1.695	0.198
5.0, 1.0	15.703	8.622	2.440	0.138
10.0, 1.0	12.849	7.043	3.183	0.080
1.5, 1.5	27.537	24.319	0.339	0.627
3.0, 1.5	24.421	20.701	1.443	1.144
5.0, 1.5	21.780	17.953	3.431	0.434
10.0, 1.5	18.268	14.453	4.836	0.202

Table 3.1: Median relative performance of routing policies E, R, J, H and O. Problems have two homogeneous stations with stochastically indistinguishable generic job classes.

From Table 3.1, note that in the main cost degradations increase with ρ, implying that the value of increased information grows with the traffic intensity. Note also that, while the medians of $\Delta(E, R)$ and $\Delta(R, J)$ both decrease as c_1 increases, for fixed ρ, the values of $\Delta(J, H)$ increase with c_1. Hence the relative cost advantage H enjoys over J grows as the job classes become more heterogeneous. Recall that J and H are approximately equal for c_1 close to 1. Finally, and very significantly, note the uniformly strong performance of H as evidenced by the small median values of $\Delta(H, O)$.

3.4.2 Stochastically Distinct Job Classes

We shall suppose, as before, that both stations are preemptive two-class $M/M/1$ systems under the $c\mu$-rule. We now fix $c_2 = \mu_2 = 1$ and allow the class 1 parameters c_1 and μ_1 to vary. We take $\rho = 0.6,\ 1.0$ and 1.5 again and compute median values of percentage cost degradations from samples of size 10 (with randomly drawn λ_1 and λ_2). The five routing

polices under investigation are as above, but with R replaced by S, the optimal static policy. The reader should note that, with $\mu_1 \neq \mu_2$, E is no longer generally optimal in the class of static polices. Theoretical considerations indicate that $C^E \geq C^S \geq C^H \geq C^O$. In fact for all 600 problems studied, it emerged that

$$C^E \geq C^S \geq C^J \geq C^H \geq C^O.$$

In an identical notation we consider the following percentage cost degradations in our study: $\Delta(E,S)$, $\Delta(S,J)$, $\Delta(J,H)$ and $\Delta(H,O)$. Tables 3.2-3.5 (respectively) contain the median values of these quantities. Each problem set is characterised by the triple (c_1, μ_1, ρ). In the layout of each table, results are grouped into a left-hand (LH) and right-hand (RH) section for each ρ-value. The LH and RH sections encompass a range of c_1 and μ_1 combinations expressing varying degrees of heterogeneity between the job classes. In the LH (resp. RH) section $c_1\mu_1 \leq c_2\mu_2$ (resp. $c_1\mu_1 \geq c_2\mu_2$). For example, the upper-left (resp. lower-right) of LH (resp. RH) correspond to problems in which the similarity between the job classes is at its greatest. As we move through either section this similarity progressively diminishes.

The results from Table 3.2 in which we report the median percentage cost degradations from E to S substantiate the findings of the numerical study of Dacre (1999). From his analyses the greatest level of (static) suboptimality is experienced for problems in which $c_1\mu_1 = c_2\mu_2$ (represented by the main diagonal in either section). As we move way from such problems this level of suboptimality decreases and in many cases E coincides with S. As previously, the cost advantage of the dynamic policies tends to grow with ρ as given by the relative performance of S with respect to J in Table 3.3. In Table 3.4 we see that the relative performance of J with respect to H strengthens in cases where the job population's characteristics are relatively similar and weakens as these differences become more pronounced. In all problem instances H outperforms J. We also note the strong performance of H in comparison with O in Table 3.5. The performance of H is particularly strong for problems in which $c_1\mu_1 = c_2\mu_2$.

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.242			2.930	1.340	0.0	0.0	10.0	0.6
0.333	0.0	1.334			2.071	0.067	0.0	5.0	
0.2	0.0	0.263	2.199			1.247	0.0	3.0	
0.1	0.0	0.0	1.438	3.113			0.203	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.325			5.730	1.948	0.033	0.0	10.0	1.0
0.333	0.0	2.345			3.833	0.104	0.0	5.0	
0.2	0.0	0.337	4.004			1.862	0.0	3.0	
0.1	0.0	0.010	2.693	5.840			0.289	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.437			10.300	2.108	0.165	0.0	10.0	1.5
0.333	0.0	3.102			5.360	0.200	0.0	5.0	
0.2	0.0	0.543	6.390			2.645	0.0	3.0	
0.1	0.0	0.396	3.035	11.330			0.403	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

Table 3.2: Relative performance of routing policies E and S: the median percentage cost degradation from policy S to policy E. Problems have two homogeneous stations and two generic job classes with $c_2 = \mu_2 = 1$.

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	24.328			19.580	20.638	21.322	19.768	10.0	0.6
0.333	21.652	22.385			21.079	22.280	20.622	5.0	
0.2	20.382	22.075	20.875			22.497	21.764	3.0	
0.1	19.387	21.179	20.410	19.285			24.381	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	39.148			29.730	32.620	34.245	31.481	10.0	1.0
0.333	34.899	35.334			32.670	35.589	33.080	5.0	
0.2	32.858	35.419	32.470			35.868	35.005	3.0	
0.1	31.239	34.138	32.100	29.450			39.205	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	61.603			43.520	52.330	54.53	51.71	10.0	1.5
0.333	56.399	55.68			51.16	56.248	53.85	5.0	
0.2	53.67	56.012	49.83			56.34	56.51	3.0	
0.1	51.45	54.39	51.27	43.33			61.662	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

Table 3.3: Relative performance of routing policies S and J: the median percentage cost degradation from policy J to policy S. Problems have two homogeneous stations and two generic job classes with $c_2 = \mu_2 = 1$.

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.655			4.477	3.708	2.689	1.341	10.0	0.6
0.333	1.166	2.103			3.193	2.423	1.218	5.0	
0.2	1.531	2.556	3.322			2.036	1.037	3.0	
0.1	1.845	2.987	3.867	4.693			0.648	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.910			7.24	5.737	3.807	2.148	10.0	1.0
0.333	1.726	2.974			4.841	3.475	2.000	5.0	
0.2	2.081	3.701	4.974			2.913	1.610	3.0	
0.1	2.509	4.260	5.926	7.461			0.904	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.758			7.13	5.338	3.024	1.484	10.0	1.5
0.333	0.913	2.564			4.343	3.144	1.168	5.0	
0.2	1.498	3.356	4.545			2.588	0.792	3.0	
0.1	1.980	3.458	5.396	7.28			0.753	1.5	
	1.5	3.0	5.0	10.0	c_1			μ_1	

Table 3.4: Relative performance of routing policies J and H: the median percentage cost degradation from policy H to policy J. Problems have two homogeneous stations and two generic job classes with $c_2 = \mu_2 = 1$.

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.001			0.0	0.002	0.030	0.002	**10.0**	**0.6**
0.333	0.001	0.002			0.0	0.020	0.001	**5.0**	
0.2	0.001	0.021	0.0			0.007	0.001	**3.0**	
0.1	0.003	0.032	0.002	0.0			0.001	**1.5**	
	1.5	**3.0**	**5.0**	**10.0**	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.002			0.003	0.067	0.204	0.077	**10.0**	**1.0**
0.333	0.020	0.025			0.022	0.105	0.053	**5.0**	
0.2	0.109	0.109	0.018			0.027	0.020	**3.0**	
0.1	0.196	0.226	0.050	0.003			0.002	**1.5**	
	1.5	**3.0**	**5.0**	**10.0**	c_1			μ_1	

μ_1			c_1	0.1	0.2	0.333	0.667		ρ
0.667	0.140			0.271	0.362	1.099	1.587	**10.0**	**1.5**
0.333	1.005	0.194			0.268	0.447	1.076	**5.0**	
0.2	1.102	0.318	0.174			0.270	0.585	**3.0**	
0.1	1.695	1.099	0.433	0.167			0.152	**1.5**	
	1.5	**3.0**	**5.0**	**10.0**	c_1			μ_1	

Table 3.5: Relative performance of routing policies H and O: the median percentage cost degradation from policy O to policy H. Problems have two homogeneous stations and two generic job classes with $c_2 = \mu_2 = 1$.

3.5 Network Design

We now return to the network design problem posed in Section 2.10. In the analysis of the routing problem our focus has been, in the main, the development of policies for routing multiple job classes to a fixed number of stations. An important question to consider is how such policies perform as the number of stations comprising the network changes. In Section 2.10 we established that, in systems comprising homogeneous stations, additional stations contribute diminishing returns under the static ESP routing regime. In the context of a simple design problem for which, in addition to the holding cost rates, each station in the network incurs a running cost we investigated what effect system parameter changes made to the optimal decision. Our interest now lies in the performance of the dynamic routing heuristic as the number of stations within the network increases (alternatively decreases). While we have no results regarding the structural properties of network systems constructed utilising the dynamic heuristic developed in this chapter we would expect that, by the improved performance of dynamic heuristics over the static heuristics investigated in the previous section, a network system constructed in which routing decisions are made according to our dynamic routing heuristic will enjoy reduced costs over an ESP system and, possibly, a reduction in the number of stations required to be built into the network for comparable performance to be achieved. In the following numerical study we investigate the benefits to the system architect in applying the dynamic routing heuristic.

3.5.1 Numerical Study

In our computational study we investigate the performance of the dynamic routing heuristic developed in this chapter and compare this with the ESP static routing heuristic for the simple design problem considered in Section 2.10. In what follows we shall use the abbreviation H to denote the dynamic routing heuristic. For this design problem the system architect has to construct a system of homogeneous stations in order to minimise

the long-run average system costs under the routing heuristics of interest. The number of stations, M, comprising the network remains to be determined. Each station m has an associated running cost per unit time, denoted K, independent of the amount of work received. In the homogeneous station set up we assume identical running costs for each station m. The system arrival rates of all job classes, the processing capabilities of each station and the cost characteristics of the job classes are considered fixed and known. In addition we also assume that the state information of the queueing job classes at each station required for the implementation of H is available.

We use Ω^H to denote the long-run average cost under H for the overall system (including holding costs and running costs), defined by

$$\Omega^H(M) = \Phi^H_{\{1,\ldots,M\}}(\boldsymbol{\lambda}^G) + MK \tag{3.34}$$

where Φ^H is used to denote the system cost function under H. The overall system cost under ESP, $\Omega^{ESP}(M)$, is given by equation (2.44). Recall that from Section 2.10.2 the function Ω^{ESP} is convex for this set up. The optimal decision under ESP is to select M station such that

$$M = \min[M : \Omega^{ESP}(M-1) \geq \Omega^{ESP}(M) \leq \Omega^{ESP}(M+1)].$$

Under H we do not have any results regarding the convexity of Ω^H and do not therefore have an equivalent simple rule for the optimal decision under H. However, it is entirely plausible that additional stations contribute diminishing returns under H. A numerical investigation could provide evidence to support this claim but we are further restricted by the limitations of state space in the DP value iteration algorithm required for the cost calculations under H. A possible way forward would be to estimate these costs via simulation. From our computational experience, a full DP value iteration approach rules out the feasibility of obtaining cost values for systems comprising four or more stations. In order to progress, problems are specifically chosen so that the optimal number of stations

(ρ, c_1, K)	λ_1	λ_2	$\Omega^{ESP}_{\mathcal{M}}$	M^*_{ESP}	$\Omega^{H}_{\{1,2\}}$	$\Omega^{H}_{\{1,2,3\}}$	M^*_H
1.0,1.5,0.5	0.201	0.799	3.108	3	2.530	2.685	2
	0.447	0.553	3.263	3	2.674	2.810	2
	0.751	0.249	3.501	3	2.892	2.975	2
1.0,3.0,0.5	0.201	0.799	3.430	3	2.850	2.986	2
	0.447	0.553	4.052	3	3.417	3.490	2
	0.751	0.249	5.005	3	4.283	4.150	3
1.0,6.0,1.0	0.201	0.799	5.116	2	4.471	5.089	2
	0.447	0.553	6.881	2	5.876	6.346	2
	0.751	0.249	9.513	3	8.049	7.995	3
1.0,6.0,1.5	0.201	0.799	6.116	2	5.471	6.589	2
	0.447	0.553	7.881	2	6.876	7.846	2
	0.751	0.249	11.013	3	9.049	9.495	2
1.3,1.5,0.5	0.201	1.099	3.902	3	3.529	3.117	3
	0.571	0.729	4.147	3	3.760	3.308	3
	1.027	0.273	4.575	3	4.185	3.577	3
1.3,1.5,1.5	0.201	1.099	6.826	2	5.529	6.117	2
	0.571	0.729	7.114	2	5.760	6.308	2
	1.027	0.273	7.575	3	6.185	6.557	2
1.3,3.0,1.0	0.201	1.099	5.725	3	4.845	4.919	2
	0.571	0.729	6.705	3	5.766	5.682	3
	1.027	0.273	8.418	3	7.448	6.756	3
1.3,3.0,1.5	0.201	1.099	7.161	2	5.845	6.419	2
	0.571	0.729	8.205	3	6.766	7.182	2
	1.027	0.273	9.918	3	8.448	8.256	3
1.3,3.0,2.5	0.201	1.099	9.161	2	7.845	9.419	2
	0.571	0.729	10.313	2	8.766	10.182	2
	1.027	0.273	12.918	3	10.448	11.256	2
1.3,6.0,1.5	0.201	1.099	7.832	2	6.480	7.023	2
	0.571	0.729	10.321	3	8.730	8.933	2
	1.027	0.273	14.603	3	12.934	11.619	3
1.3,6.0,2.5	0.201	1.099	9.832	2	8.480	10.023	2
	0.571	0.729	12.711	2	10.730	11.933	2
	1.027	0.273	17.603	3	14.934	14.619	3
1.3,6.0,3.5	0.201	1.099	11.832	2	10.480	13.023	2
	0.571	0.729	14.711	2	12.730	14.933	2
	1.027	0.273	20.603	3	16.934	17.619	2

Table 3.6: Comparison of the performances routing polices ESP and H. Stations are homogeneous with identical running costs and stochastically indistinguishable job classes. $c_2 = 1$.

under ESP is at most three. We then obtain costs for systems comprising two and three stations under H and compare these results with those for the optimal decision under ESP.

In the study we suppose that the stations are modelled as preemptive two-class $M/M/1$ systems with local scheduling according to the $c\mu$-rule. We shall only consider the special case in which the job classes are stochastically indistinguishable and take $\mu_1 = \mu_2 = 1$ for the exponential service rates of the two job classes at all stations. Recall that, by Theorem 20, ESP is the optimal static routing policy in this case. We take $c_2 = 1$ and $c_1 \geq 1$. Under the $c\mu$-rule class 1 will have higher priority in all problems considered here. The calculations required to compute the overall system cost under ESP and hence the optimal number of stations in the network are straightforward. An appropriate form of DP value iteration is used to compute the system costs under H and, via (3.34), the overall system costs. Each problem set is characterised by the triple (ρ, c_1, K) and, given the comments above, is specifically chosen to encompass problems in which the optimal number of stations under ESP is at most three. We must note that while we are restricted in the size of the systems that we can consider these problems are of sufficient complexity to yield some insight.

In Table 3.6 we compare the optimal number of stations and system performance under ESP with the system performance under H in which the network comprises two and three stations. We use M^*_{ESP} to denote the optimal number of stations under ESP routing and M^*_H to denote the number of stations used in the network under the dynamic heuristic H that realises the smallest overall system cost of the networks considered. For each problem set we report a sample of results that are typical of our findings, these examples range from a low to high class 1 arrival rate (respectively high to low class 2 arrival rate). We can see that, as expected, substantial cost savings are made via the use of H. This does not always correspond to a reduction in the number of stations utilised in the network. It is important to note that in all 650 problems studied the overall system costs for a two station network under H are less than the overall system costs for an optimal

network under ESP. The findings suggest that the system architect would be able to achieve improved system performance (or at worst, a similar performance) utilising fewer stations under the dynamic heuristic routing policy for the problem set ups considered here.

3.6 Conclusion

We described an approach to the development of dynamic routing policies based on a single policy improvement step applied to an optimal static policy which results in simply structured dynamic routing polices. The heuristics generalise the "join the shortest queue" policy in a way appropriate to the multi-class context. A numerical study provides evidence of the very strong performance of the routing heuristics. In a network design problem we showed that when constructing networks of homogeneous stations application of these routing heuristics results in reduced system costs in comparison with a static routing regime. The numerical results suggest that, in such problems, it is possible improve system costs utilising fewer machines under the dynamic routing heuristics.

Chapter 4

Index Policies for the Routing of Background Jobs

4.1 Introduction

In this chapter we develop an alternative approach to the development of dynamic routing policies. Whittle (1996) and Niño-Mora (2002) have proposed the use of methodologies related to the class of restless bandit problems (RBPs)to solve routing problems. This class of problems are now understood to be intractable and Papadimitriou and Tsitsiklis (1999) showed RBPs to be PSPACE-hard. Whittle's (1988) approach to the analysis of RBPs used a Lagrangian relaxation of the original problem to develop an *index policy* for RBPs. This index policy will not in general be optimal. Weber and Weiss (1990,1991) have proved the asymptotic optimally of Whittle's index policy under given conditions. However, Whittle's proposed index is only defined for projects which pass a test of *indexability.* This requirement can be very difficult to establish and may not hold, although Niño-Mora (2001,2002) has identified sufficient conditions for project indexability based on the achievable region approach. A developing body of evidence testifies to the strong performance of such index policies in a range of application contexts. See, for example, Ansell, Glazebrook, Niño-Mora and O'Keeffe (2003), Glazebrook and Mitchell (2004) and Glazebrook, Niño-Mora and Ansell (2002).

In this chapter we describe a mathematical model for our routing problem and describe an approach to the development of index policies by taking an approximative approach to

to Whittle's proposal for indexability. The approximative scheme for the development of a station index is implemented via a policy improvement approach. The station index is a measure of its degree of congestion and are shown to be increasing and non-linear in the workload. The index policy will route an arriving generic job to the station of smallest index. We conclude the chapter with a numerical investigation into the performance of such routing policies.

4.2 The Dynamic Routing Problem

Our routing problem of interest remains broadly that of the dynamic routing problem developed in Chapter 3. We shall reintroduce the dynamic routing problem which, in the current context, utilises local scheduling policies allowing for certain priority orderings amongst the competing job classes. Our problem concerns a distributed system that comprises a central controller and a set of stations $\mathcal{M} = \{1, 2, \ldots, M\}$. Jobs from a number of distinct classes arrive at the system for processing. Job classes (and their constituent jobs) are either dedicated or generic. We assume that dedicated jobs arrive directly at their specified station for processing while the choice of station for generic jobs remains open. We use D_m to denote the class of jobs dedicated to station $m \in \mathcal{M}$ and G to denote the class of generic jobs. Hence $E = G \cup D_1 \cup \ldots \cup D_M$ is the set of jobs allowed access to the system while $E_m = G \cup D_m$ is the set of job classes allowed access to station $m \in \mathcal{M}$. Jobs arrive in independent Poisson streams. Generic jobs arrive at the system with rate λ. Those dedicated to station m arrive there at rate λ_{dm} for D_m, $m \in \mathcal{M}$. The *routing problem* concerns the choice of station to process the generic jobs to minimise some measure of system holding cost. We formulate this as a decision problem as follows:

(i) We write $\mathbf{N}(t) = \{\mathbf{N}_m(t)\}_{m \in \mathcal{M}}$ for the *state of the system* at time $t \in \mathbb{R}^+$, with $\mathbf{N}_m(t)$ for the *state of station m*. The latter is given by $\mathbf{N}_m(t) = \{N_{dm}(t), N_{gm}(t)\}$, namely the two-vector which is the number of dedicated and generic jobs respectively at station m at time t;

(ii) The *decision epochs* for the routing problem are the times at which generic jobs arrive at the system and will be the event times of a Poisson process with rate λ. At each decision epoch, the actions available to the system controller are $A = \{a_1, a_2, \ldots, a_M\}$, where a_m denotes the routing of the newly arrived job to station m, thus increasing the generic queue length there by one;

(iii) Between successive decision epochs each station m evolves (via the arrival of dedicated jobs from D_m and via service completions) as a two-class $M/M/1$ system as follows: dedicated jobs arrive from D_m according to a Poisson stream with rate λ_{dm}. A single server implements some admissible scheduling policy $u_m \in \mathcal{U}_m$ for which we assume that the dedicated jobs have (preemptive) priority over generic jobs. We denote this scheduling policy $D_m \rightarrow G$. Under this policy for scheduling work at the station generic jobs have the role of background traffic that can only access processing if no dedicated jobs are present. The service requirements for all jobs at station m are exponentially distributed, with μ_{dm} the rate for dedicated jobs and ν_m the rate for generic jobs. We assume that all arrival and service processes are mutually independent;

(iv) A holding cost rate $c_{dm} \geq 0$ is associated with dedicated jobs at station m with $c_g \geq 0$ the cost rate for generic jobs. Given the scheduling policy $D_m \rightarrow G$, $m \in \mathcal{M}$, our goal is to develop routing policies to minimise the long run average cost rate

$$E\left(\sum_{m\in\mathcal{M}} c_{dm}N_{dm} + c_g \sum_{m\in\mathcal{M}} N_{gm}\right) \tag{4.1}$$

where the expectation in (4.1) is taken in steady state. However, under the scheduling policy $D_m \rightarrow G$ the total cost due to dedicated jobs is independent of the routing policy. For the problem of determining the best routing policy we may, without loss of generality, suppose that $c_{dm} = 0$, $m \in \mathcal{M}$ and $c_g = 1$. Hence our objective in (4.1) is equivalent to

the minimisation of

$$E\left(\sum_{m\in\mathcal{M}} N_{gm}\right), \tag{4.2}$$

the expected number of generic jobs in the system. Note that we assume that the system is stable in that a steady state solution with finite queue lengths exists.

As discussed in Chapter 3, the problem of determining an optimal routing policy can be formulated as a stochastic dynamic programme (DP). We noted that, for problems of a reasonable size, DP will generally be an ineffective tool of analysis. In addition, theoretical difficulties posed by single class systems with homogeneous servers suggest that any quest for simple closed form optimal policy is unlikely to be fruitful. Our aim is to develop heuristic policies of simple structure achieving good overall performance.

In Chapter 3 the dynamic routing heuristic takes the form of an *index policy*, at each decision epoch t the destination of the arriving generic job is chosen by the evaluation of some calibrating value, known as *the index*, for each station m which is dependent upon the current state of station m only. By Theorem 25 these key index values are congestion measures of the station concerned. In its application, the dynamic routing heuristic would route the incoming generic job to the station with the smallest index. It should be noted that the dynamic routing heuristic is constructed in two stages. The first stage involves the solution of the full routing problem in determination of the optimal static policy, which for this problem can be found by applying the methods described in Chapter 2. The dynamic routing heuristic is then obtained by the application of a single policy improvement step to the optimal static policy.

In what follows, we develop an alternative approach to station indexation in which we deploy a prescription of Whittle (1988) developed for RBPs. Again the index developed here reflects the degree of congestion of each particular station and in its application the index policy will route the incoming generic job to the station with the smallest index. By taking an approximative approach to Whittle's proposal we develop a simple form of

index which, importantly, is *decomposable* in the sense that it concerns individual stations only and in no way makes any appeal to the routing problem in its entirety.

4.3 Indices for Service Stations

In following Whittle's (1988) prescription for the development of an index for station m, we focus on *station m alone* and ignore other aspects of our (multi-station) routing problem. As a consequence we can drop the station suffix until the conclusion of Section 4.4. In particular, we consider a class of admission control problems involving a single station which are defined as follows:

(I) Dedicated jobs from class D and generic jobs from class G arrive at a single server according to independent Poisson streams with rates λ_d and λ respectively. All jobs have exponentially distributed processing requirements, with μ the rate for dedicated jobs and ν the rate for generic jobs. Note that parameters λ_d, μ and ν will be those specific to the station in the multi-station problem of Section 4.2 and λ is the generic arrival rate for the system. All arrival, service and feedback processes are independent and the server implements the (preemptive) priority policy $D \rightarrow G$ for scheduling the work at the station. Hence, generic jobs are served *only* when no dedicated jobs are present;

(II) All dedicated jobs arrive directly at the station and will eventually receive service. Each arriving generic job is subject to the admission control policy which may admit the job into the station (accept) or alternatively refuse entry into the station (reject). In the full (multi-station) problem these two actions correspond to routing the generic stream towards this station (accept) and away from this station (reject). In addition to incurring holding costs of unit rate for generic jobs as in (4.2), a cost equal to W is incurred on every occasion that an arriving generic job is rejected;

(III) If we let π denote a general deterministic, stationary and Markov policy for deciding whether to admit arriving generic jobs, then the total expected cost incurred

under policy π is written

$$C^{\pi}(W) = E_{\pi}\{N_g + WI(N_d, N_g)\} \tag{4.3}$$

In (4.3), N_d and N_g are respectively the number of dedicated and generic jobs in the system, E_{π} is an expectation taken with respect to the steady state distribution of (N_d, N_g) under policy π and I is an indicator such that $I(n_d, n_g)$ is 1 if an arriving generic job is rejected when the system is in state (n_d, n_g) and is 0 otherwise. We also write

$$C^{OPT}(W) = \min_{\pi} C^{\pi}(W)$$

for the minimum cost associated with the single station problem with rejection charge W. We further introduce $A(W)$ as the set of states in which an optimal policy chooses to accept an arriving generic job when the rejection charge is W. The following applies Whittle's (1988) notion of indexability to the current problem context.

Definition 9

If $A(W)$ is increasing in W, i.e.

$$W_1 > W_2 \Rightarrow A(W_1) \supseteq A(W_2)$$

then we say that the station is indexable.

If a station has the indexability property, we are then able to define the index for a given state (n_d, n_g) as the rejection penalty which renders both actions accept/reject optimal when the system is in the given state. The station index can be thought of as a *fair charge* for rejecting an incoming generic job when the station is in state (n_d, n_g). Small index values represent a greater willingness to accepts jobs in the full multi-station problem. We would expect good policies to utilise stations with small index values.

Definition 10

For an indexable station, the index function $W : \mathbb{N}^2 \to \mathbb{R}$ *is defined by*

$$W'(n_d, n_g) = \inf\{W; (n_d, n_g) \in A(W)\}.$$

$W'(n_d, n_g)$ *is called the* index for state (n_d, n_g).

Under indexability, an *index policy* for the multi-station routing problem of Section 4.2 will always choose the service station with the smallest current index value. However, establishing station indexability is exceptionally difficult and success has only been achieved for objects with countable (or finite) state spaces in one dimension. Our single station admission control problem has a two-dimensional state space. Theoretical difficulties arise from the fact that no natural ordering of the states with respect to index values can easily form the basis of an analysis. In the current context we have no proof of (exact) indexability and no derived index function. One could approach the verification of indexability from a purely computational perspective. However, such an approach still poses a formidable challenge. In addition, we would have to question whether a purely numerical approach could provide any insight. Hence we shall proceed in an *approximative* fashion by developing good (though, in general, suboptimal) policies for the above single station admission control problem with rejection charge W which have the property that the corresponding acceptance sets $\hat{A}(W)$, say, are increasing in W as required by Definition 9. The corresponding index function $\hat{W} : \mathbb{N}^2 \to \mathbb{R}$ will then be given by

$$\hat{W}'(n_d, n_g) = \inf\{W' : (n_d, n_g) \in \hat{A}(W)\} \tag{4.4}$$

as in Definition 10.

In the next section we develop the $\hat{A}(W')$ by the application of a single policy improvement step to an optimal static policy for the above single station admission control problem with rejection charge W'

4.4 Developing Approximate Indices for Service Stations

We begin our analysis of the (single station) admission control problem with rejection charge W' described in Section 4.3 by focusing on the class of *static policies* in which each arriving generic job is accepted into the system with some probability $p \in [0,1]$. The resulting stream of generic jobs accepted into the system is Poisson with rate λp. Following (4.3) we write the corresponding system cost (in an obvious notation) as

$$C^p(W') = E_p(N_g) + W'\lambda(1-p). \tag{4.5}$$

A closed form formula for the expectation in (4.5) is available. Our two-class $M/M/1$ system model for the station satisfies the Generalised Conservation Laws (GCL) of Section 2.5. For the system to be stable under static policy p we must have

$$\rho_d + p\rho_g < 1, \tag{4.6}$$

where in (4.6) we have used the notation $\rho_d = \lambda_d/\mu$ and $\rho_g = \lambda/\nu$. Then by Definition 4 the following equations are satisfied for scheduling rule $D \rightarrow G$:

$$\frac{E(N_d)}{\mu} = \frac{\rho_d \mu^{-1}}{1-\rho_d} \tag{4.7}$$

and

$$\frac{E(N_d)}{\mu} + \frac{E_p(N_g)}{\nu} = \frac{\rho_d \mu^{-1} + p\rho_g \nu^{-1}}{1-\rho_d - p\rho_g}. \tag{4.8}$$

From (4.5), (4.7) and (4.8) we deduce that $C^p(W')$ is given as follows when $\rho_d + p\rho_g < 1$:

$$C^p(W') = \lambda p\{\rho_d \mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-1}(1-\rho_d-\rho_g p)^{-1} + W'\lambda(1-p). \tag{4.9}$$

We now write $p(W)$ for the value of p which minimises $C^p(W)$. $p(W)$ determines the optimal static policy for the problem. Lemma 27 follows simply from (4.9). In the statement of the result we use the notation $(x)_0^1$ when $x \in \mathbb{R}$ as follows:

$$(x)_0^1 = \begin{cases} 0, & x < 0, \\ x, & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$$

Lemma 27

The optimal static policy for the admission control problem with rejection charge W is given by

$$p(W) = \left(\rho_g^{-1} \left[1 - \rho_d - \sqrt{\frac{\{\rho_d \mu^{-1} + (1-\rho_d)\nu^{-1}\}}{W}} \right] \right)_0^1 \tag{4.10}$$

We define the quantities $\underline{W}$ and $\overline{W}$ by

$$\underline{W} = \sup\{W;\ p(W) = 0\}$$

and

$$\overline{W} = \inf\{W;\ p(W) = 1\}.$$

It is trivial from (4.10) that

$$\underline{W} = \{\rho_d \mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-2} \tag{4.11}$$

and

$$\overline{W} = \{\rho_d \mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d-\rho_g)^{-2} \quad \text{when} \quad \rho_d + \rho_g < 1.$$

If $\rho_d + \rho_g > 1$, then there are no W-values for which $p(W) = 1$ and we write $\overline{W} = \infty$.

We now apply a single policy improvement step to the optimal static policy $p(W)$ to develop a dynamic policy for the single station admission control problem with rejection charge W. It is this policy and its associated acceptance sets $\hat{A}(W)$ which will yield our indices $\hat{W}$ as in (4.4). We follow the approach to policy improvement for average cost decision problems proposed by Tijms (1994). It is this same approach that was applied successfully in the development of the dynamic routing heuristic for multi-class dynamic routing problems of the previous chapter.

Our interest lies in the quantities $\Delta(n_d, n_g, W)$ defined as follows:

Consider a situation in which there is a new generic job arrival in state (n_d, n_g) for the single station admission control problem with rejection charge W described by (I)-(III) above. We define the quantity $\Delta(n_d, n_g, W)$ as the difference in total expected costs over an infinite horizon between the policy which admits the generic job arrival and thereafter operates static policy $p(W)$ and the policy which rejects the generic arrival (incurring charge W) and thereafter operates static policy $p(W)$.

The region of acceptance is then given by

$$\hat{A}(W) = \{(n_d, n_g);\ \Delta(n_d, n_g, W) \leq 0\}. \tag{4.12}$$

Central to the policy improvement approach are the state-valued quantities called relative costs (alternatively biases) associated with a stochastic system under a given policy. We can express the quantities $\Delta(n_d, n_g, W)$ in terms of the relative costs for the system. We use $\hat{h}(n_d, n_g)$ to denote the relative cost associated with system state (n_d, n_g) when the single station with rejection charge W has the admission of generic arrivals controlled by optimal static policy $p(W)$. We utilise standard MDP theory, together with the fact that our single station regenerates upon every return to the empty state to

develop an expression for $\hat{h}(n_d, n_g)$ as follows:

$$\hat{h}(n_d, n_g) = \hat{C}_{p(W)}(n_d, n_g) - [C^{p(W)} - W\lambda\{1 - p(W)\}]\hat{T}_{p(W)}(n_d, n_g). \tag{4.13}$$

In (4.13), $\hat{C}_{p(W)}(n_d, n_g)$ is the total expected cost incurred by waiting generic jobs at the station operating under optimal static policy $p(W)$ from initial state (n_d, n_g) until the station first enters state (0,0) (i.e. until it first becomes empty) and $\hat{T}_{p(W)}(n_d, n_g)$ is the corresponding expected time. Note that $C^{p(W)}(W)$ is given by the expression in (4.9) with $p = p(W)$.

We provide an alternative characterisation of the relative costs in (4.13). In what follows we shall need to consider the system state at all event epochs. We write $\Lambda = \lambda_d + \mu + \lambda + \nu$ and pursue a uniformisation approach in which event epochs occur as a Poisson process with rate Λ, with "virtual" events (i.e. corresponding to service completions within a job class not being served) resulting in transitions from the current system state to itself. Consider state (n_d, n_g) with $n_d \neq 0$. Under the preemptive priority policy $D \rightarrow G$, job class D will be chosen for service in this state. We then have that

$$\begin{aligned} C^{p(W)}(W) + \Lambda\hat{h}(n_d, n_g) = n_g + \lambda p(W)\hat{h}(n_d, n_g + 1) + \lambda\{1 - p(W)\}\{\hat{h}(n_d, n_g) + W\} \\ + \lambda_d\hat{h}(n_d + 1, n_g) + \nu\hat{h}(n_d, n_g) + \mu\hat{h}(n_d - 1, n_g) \end{aligned} \tag{4.14}$$

Equations for the states $(0, n_g)$ can be developed similarly. The relative costs in (4.13) satisfy these equations and the additional requirement that $\hat{h}(0, 0) = 0$.

Our dynamic policy obtained via the application of a single policy improvement step to optimal static policy $p(W)$ will choose between the actions {accept incoming generic job,reject incoming generic job} in state (n_d, n_g) to achieve the minimum in the following

expression:

$$n_g + \lambda \min\{\hat{h}(n_d, n_g + 1), \hat{h}(n_d, n_g) + W\} + \lambda_d \hat{h}(n_d + 1, n_g) + \nu \hat{h}(n_d, n_g) + \mu \hat{h}(n_d - 1, n_g) \quad (4.15)$$

and similarly for states of the form $(0, n_g)$. The first term within the minimisation in (4.15) corresponds to the action "accept", while the second term corresponds to the action "reject".

Tijms (1994) provided an economic interpretation of the relative costs by which the quantity $\hat{h}(n_d, n_g + 1) - \hat{h}(n_d, n_g)$ may be understood as the difference in total expected costs (holding costs and rejection penalties) incurred at the single station under optimal static admission control policy $p(W)$ over an infinite horizon between starting in state $(n_d, n_g + 1)$ and state (n_d, n_g). Under this interpretation of the relative costs we can for our single station admission control problem deduce that the quantities $\Delta(n_d, n_g, W)$ are given by

$$\Delta(n_d, n_g, W) = \hat{h}(n_d, n_g + 1) - \{\hat{h}(n_d, n_g) + W\}.$$

Hence we can re-express (4.12) as

$$\hat{A}(W) = \{(n_d, n_g); W \geq \hat{h}(n_d, n_g + 1) - \hat{h}(n_d, n_g)\}. \quad (4.16)$$

where, from (4.13), we have that

$$\hat{h}(n_d, n_g + 1) - \hat{h}(n_d, n_g) = \hat{C}_{p(W)}(n_d, n_g + 1) - \hat{C}_{p(W)}(n_d, n_g) - [C^{p(W)}(W) - W\lambda\{1 - p(W)\}]\{\hat{T}_{p(W)}(n_d, n_g + 1) - \hat{T}_{p(W)}(n_d, n_g)\}. \quad (4.17)$$

We now provide an explicit characterisation of the acceptance region $\hat{A}(W)$ in Lemma 28.

Lemma 28

The policy for the admission control problem with rejection charge W obtained by applying a policy improvement step to the optimal static policy has acceptance region

$$\hat{A}(W) = [(n_d, n_g); W \geq \{n_d\mu^{-1} + (n_g + 1)\nu^{-1}\}\{1 - \rho_d - \rho_g p(W)\}^{-1}].$$

Proof

We shall prove the result by utilising (4.16) and (4.17). In order to make use of (4.17) we shall require formulae for $\hat{T}_{p(W)}(n_d, n_g)$ and $\hat{C}_{p(W)}(n_d, n_g + 1) - \hat{C}_{p(W)}(n_d, n_g)$. We now sketch how these are obtained. In order to analyse $\hat{T}_{p(W)}(n_d, n_g)$, we shall first need $\hat{T}^D(n_d)$, defined to be the expected time required to empty the station of *dedicated traffic* (only) from initial dedicated state n_d.

By a simple argument based on busy periods of the process of dedicated traffic, we have that

$$\lambda_d \hat{T}^D(1) = \rho_d(1 - \rho_d)^{-1}. \tag{4.18}$$

Further, the nature of the dedicated process implies that

$$\hat{T}^D(n_d) = n_d \hat{T}^D(1). \tag{4.19}$$

From (4.18) and (4.19) we conclude that

$$\hat{T}^D(n_d) = n_d\mu^{-1}(1 - \rho_d)^{-1}. \tag{4.20}$$

A similar argument which utilises the fact that the station must be emptied of all dedicated traffic before any generic processing can begin yields that

$$\hat{T}_{p(W)}(n_d, n_g) = \hat{T}^D(n_d) + \{n_g + \lambda p(W)\hat{T}^D(n_d)\}\hat{T}_{p(W)}(0, 1). \tag{4.21}$$

However, a one-step value iteration argument gives

$$\{\lambda_d + \lambda p(W) + \nu\}\, \hat{T}_{p(W)}(0,1) = 1 + \lambda_d \hat{T}_{p(W)}(1,1) + \lambda p(W) \hat{T}_{p(W)}(0,2). \tag{4.22}$$

Utilising a similar argument to that used to obtain (4.19), we have that $\hat{T}_{p(W)}(0,2)$ in the third term on the r.h.s. of (4.22) is given by

$$\hat{T}_{p(W)}(0,2) = 2\hat{T}_{p(W)}(0,1). \tag{4.23}$$

From (4.20)-(4.23) we conclude that

$$\hat{T}_{p(W)}(n_d, n_g) = \left\{n_d \mu^{-1} + n_g \nu^{-1}\right\} \left\{1 - \rho_d - \rho_g p(W)\right\}^{-1}. \tag{4.24}$$

We now proceed to consider $\hat{C}_{p(W)}(n_d, n_g + 1) - \hat{C}_{p(W)}(n_d, n_g)$. A simple argument based on an excess cost rate of one for the $(n_d, n_g + 1)$ case during the initial period of emptying the station of dedicated traffic yields that

$$\begin{aligned} &\hat{C}_{p(W)}(n_d, n_g + 1) - \hat{C}_{p(W)}(n_d, n_g) \\ &\qquad = \hat{T}^D(n_d) + E_X\{\hat{C}_{p(W)}(0, n_g + 1 + X) - \hat{C}_{p(W)}(0, n_g + X)\}, \end{aligned} \tag{4.25}$$

where X denotes a random number of generic arrivals during this initial period and E_X is an expectation taken with respect to the distribution of X. From (4.20) and (4.25) it will be enough for us to analyse $\hat{C}_{p(W)}(0, N + 1) - \hat{C}_{p(W)}(0, N)$ for arbitrarily chosen non-negative integer N. An argument similar to that which yielded (4.25) gives

$$\begin{aligned} \hat{C}_{p(W)}(0, N + 1) - \hat{C}_{p(W)}(0, N) &= \hat{T}_{p(W)}(0, N) + \hat{C}_{p(W)}(0, 1) \\ &= N\nu^{-1}\{1 - \rho_d - \rho_g p(W)\}^{-1} + \hat{C}_{p(W)}(0, 1), \end{aligned} \tag{4.26}$$

where we have used (4.24) to obtain (4.26).

Now consider the system evolving from initial state $(0,1)$. If the first random event is an arrival, then we may assume w.l.o.g. that the single generic job present initially is laid aside (incurring costs at rate one) while a busy period of the dedicated/generic system takes place. Once this busy period has terminated, the system will be in state $(0,1)$ again. Hence it is easy to see via an argument which conditions on the nature of the first random event, that

$$\{\lambda_d + \lambda p(W) + \nu\}\hat{C}_{p(W)}(0,1)$$
$$= 1 + \{\lambda_d + \lambda p(W)\}\{\bar{C}_{p(W)} + \bar{T}_{p(W)} + \hat{C}_{p(W)}(0,1)\}, \qquad (4.27)$$

where in (4.27), $\bar{C}_{p(W)}$ and $\bar{T}_{p(W)}$ denote respectively the expected cost incurred during a busy period of the full dedicated/generic system and its expected duration. A simple argument based on such busy periods yields that

$$\bar{C}_{p(W)} = [C^{p(W)}(W) - W\lambda\{1-p(W)\}]\{\lambda_d + \lambda p(W)\}^{-1}\{1-\rho_d-\rho_g p(W)\}^{-1} \qquad (4.28)$$

and

$$\bar{T}_{p(W)} = \{\lambda_d + \lambda p(W)\}^{-1}\{\rho_d + \rho_g p(W)\}\{1-\rho_d-\rho_g p(W)\}^{-1}. \qquad (4.29)$$

We then infer from (4.27)-(4.29) that

$$\hat{C}_{p(W)}(0,1) = \nu^{-1}[1 + C^{p(W)}(W) - W\lambda\{1-p(W)\}]\{1-\rho_d-\rho_g p(W)\}^{-1},$$

whence from (4.25) and (4.26)

$$\hat{C}_{p(W)}(n_d, n_g+1) - \hat{C}_{p(W)}(n_d, n_g) = \hat{T}^D(n_d)$$
$$+ \nu^{-1}[n_g + 1 + E(X) + C^{p(W)}(W) - W\lambda\{1-p(W)\}]\{1-\rho_d-\rho_g p(W)\}^{-1}, \qquad (4.30)$$

where

$$E(X) = \lambda p(W)\hat{T}^D(n_d) = \lambda p(W) n_d \mu^{-1}(1-\rho_d)^{-1}. \tag{4.31}$$

It now follows easily from (4.20), (4.24), (4.30) and (4.31) that

$$\begin{aligned}\hat{C}_{p(W)}(n_d, n_g+1) - \hat{C}_{p(W)}(n_d, n_g) \\ - [C^{p(W)}(W) - W\lambda\{1-p(W)\}]\{\hat{T}_{p(W)}(n_d, n_g+1) - \hat{T}_{p(W)}(n_d, n_g)\} \\ = \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}\left\{1-\rho_d-\rho_g p(W)\right\}^{-1}. \end{aligned} \tag{4.32}$$

The result is an immediate consequence of (4.16), (4.17) and (4.32). □

Corollary 29

$\hat{A}(W)$ is increasing in W.

Proof

Suppose that $\rho_d + \rho_g < 1$. The complementary case is dealt with similarly. From Lemmas 27 and 28 we have that

$$\hat{A}(W) = \left[(n_d, n_g);\ W \geq \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}(1-\rho_d)^{-1}\right],\ W \leq \underline{W}, \tag{4.33}$$

and

$$\hat{A}(W) = \left[(n_d, n_g);\ W \geq \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}(1-\rho_d-\rho_g)^{-1}\right],\ W \geq \overline{W}. \tag{4.34}$$

It is evident from (4.33) and (4.34) that $\hat{A}(W)$ is increasing in the ranges $W \in [0, \underline{W}]$ and $W \in [\overline{W}, \infty)$. It remains to show that

$$\underline{W} \leq W_1 \leq W_2 \leq \overline{W} \Rightarrow \hat{A}(\underline{W}) \subseteq \hat{A}(W_1) \subseteq \hat{A}(W_2) \subseteq \hat{A}(\overline{W}).$$

First, suppose that $(n_d, n_g) \in \hat{A}(\underline{W})$ and $\underline{W} \leq W_1 \leq \overline{W}$. It must follow from (4.33) that

$$\underline{W} \geq \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}(1-\rho_d)^{-1}$$

$$\Rightarrow \underline{W}(1-\rho_d)\{1-\rho_d-\rho_g p(W_1)\}^{-1}$$

$$\geq \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}\{1-\rho_d-\rho_g p(W_1)\}^{-1}. \tag{4.35}$$

But, from Lemma 27, observe that

$$1-\rho_d-\rho_g p(W_1) = \sqrt{\frac{\{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}}{W_1}}$$

and hence the l.h.s. of (4.35) becomes

$$\underline{W}\sqrt{W_1}(1-\rho_d)[\{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}]^{-\frac{1}{2}} = \sqrt{\underline{W}W_1} \leq W_1, \tag{4.36}$$

using (4.11). Combining (4.35) and (4.36) we deduce that

$$\begin{aligned}(n_d, n_g) \in \hat{A}(\underline{W}) &\Rightarrow \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}\{1-\rho_d-\rho_g p(W_1)\}^{-1} \leq W_1 \\ &\Rightarrow (n_d, n_g) \in \hat{A}(W_1)\end{aligned} \tag{4.37}$$

by Lemma 28. From (4.37) it follows that $\hat{A}(\underline{W}) \subseteq \hat{A}(W_1)$, as required. The cases $\hat{A}(W_1) \subseteq \hat{A}(W_2)$ and $\hat{A}(W_2) \subseteq \hat{A}(\overline{W})$ are dealt with similarly. □

In the light of Corollary 29, we can introduce the index function $\hat{W} : \mathbb{N}^2 \to \mathbb{R}$ as in (4.4). The following result gives this index in closed form.

Theorem 30

(a) If $\rho_d + \rho_g \leq 1$, the function $\hat{W} : \mathbb{N}^2 \to \mathbb{R}$ is given by $\hat{W}(n_d, n_g) = f_1\{n_d\mu^{-1} + (n_g +$

$1)\nu^{-1}\}$, *where* $f_1 : \mathbb{R}^+ \to \mathbb{R}^+$ *is such that*

$$f_1(x) = \begin{cases} x(1-\rho_d)^{-1}, & x \leq \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-1}, \\ x^2\{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}^{-1}, & \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-1} \leq x \\ & \leq \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d-\rho_g)^{-1}, \\ x(1-\rho_d-\rho_g)^{-1}, & x \geq \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d-\rho_g)^{-1}. \end{cases}$$

(b) If $\rho_d + \rho_g \geq 1$, *the function* $\hat{W} : \mathbb{N}^2 \to \mathbb{R}$ *is given by* $\hat{W}(n_d, n_g) = f_2\{n_d\mu^{-1} + (n_g + 1)\nu^{-1}\}$, *where* $f_2 : \mathbb{R}^+ \to \mathbb{R}^+$ *is such that*

$$f_2(x) = \begin{cases} x(1-\rho_d)^{-1}, & x \leq \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-1}, \\ x^2\{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}^{-1}, & x \geq \{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\}(1-\rho_d)^{-1}. \end{cases}$$

Proof

We consider case (a). Suppose that $(n_d, n_g) \in \hat{A}(\underline{W})$. It must follow that

$$0 \leq \hat{W}(n_d, n_g) \leq \underline{W}.$$

From (4.4) and (4.33)we deduce that

$$\hat{W}(n_d, n_g) = \left\{n_d\mu^{-1} + (n_g + 1)\nu^{-1}\right\}(1-\rho_d)^{-1}. \tag{4.38}$$

If we now suppose that $(n_d, n_g) \notin \hat{A}(\underline{W})$ and $(n_d, n_g) \in \hat{A}(\overline{W})$ it will then follow from (4.4) that

$$\underline{W} < \hat{W}(n_d, n_g) \leq \overline{W}.$$

From (4.4) and Lemma 28, we deduce that

$$\hat{W}(n_d, n_g) = \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}\left[1 - \rho_d - \rho_g p\{\hat{W}(n_d, n_g)\}\right]^{-1}. \tag{4.39}$$

Combining (4.39) with Lemma 27, it follows that

$$\hat{W}(n_d, n_g) = \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}^2\left\{\rho_d\mu^{-1} + (1-\rho_d)\nu^{-1}\right\}^{-1} \tag{4.40}$$

for this case.

Finally, suppose that $(n_d, n_g) \notin \hat{A}(\overline{W})$ and hence that

$$\hat{W}(n_d, n_g) \geq \overline{W}.$$

From (4.4) and (4.34) we conclude that

$$\hat{W}(n_d, n_g) = \left\{n_d\mu^{-1} + (n_g+1)\nu^{-1}\right\}(1 - \rho_d - \rho_g)^{-1}. \tag{4.41}$$

Theorem 30(a) now follows from (4.38), (4.40) and (4.41). Theorem 30(b) is dealt with similarly. □

As is apparent from Theorem 30 the index $\hat{W}(n_d, n_g)$ is a function of the quantity $\omega(n_d, n_g) = n_d\mu^{-1} + (n_g+1)\nu^{-1}$ which has the interpretation as the current total workload at the station *following* the admission of the newly arrived generic job. The following result follows easily from Theorem 30.

Corollary 31

The index $\hat{W} : \mathbb{N}^2 \to \mathbb{R}$ *is an increasing function of the workload* $\omega : \mathbb{N}^2 \to \mathbb{R}$.

Now recall the multi-station problem of Section 4.2 and restore the station subscript m. Hence $\hat{W}_m(n_{dm}, n_{gm})$ is the station m index for state (n_{dm}, n_{gm}). The derived index policy will route a generic job arriving at time $t \in \mathbb{R}^+$ to whichever station has the smallest

value of current index $\hat{W}_m\{N_{dm}(t), N_{gm}(t)\}$. It is easy to see from Theorem 30 that in the special case of *homogeneous stations* (i.e., both the stochastic dynamics of the dedicated traffic and ν_m are the same for all m) the index policy chooses the station with smallest workload $\omega_m\{N_{dm}(t), N_{gm}(t)\}$.

4.4.1 Extension to the Klimov Network Model

In the analysis above we have focused on the case in which each station is modelled as a two-class $M/M/1$ queueing system admitting a single dedicated job class and a single generic job class. This simple model reduces the inherent complexities of the routing problem, allowing for the clear development of the index policy. We are not restricted to such cases and we may consider more complex models involving multiple dedicated job classes. The extension to these more general models is possible due to the nature of the priority scheduling policy in force at each station in which dedicated traffic is given (preemptive) priority over the generic work. Queueing generic jobs may only access processing if no dedicated jobs are present.

One such example is a problem in which each station is modelled as a Klimov network (see Example 1, Section 2.2) with a single generic job class and multiple dedicated job classes. This is a considerably more complex problem and we refer the reader to Glazebrook and Kirkbride (2004) for a thorough analysis. The analysis above continues to hold when the stations are modelled as Klimov networks and follows the same approximative approach to station indexation (albeit rather more involved). The acceptance sets for the associated single station admission control problems are (again) increasing in the rejection charge W as required by Definition 9. The resulting index function gives the station indices in closed form. This index function is increasing in the workload, hence, the index policy will route an arriving generic job to the station with currently smallest index value.

$\bar{\rho}$	ν	λ_d	λ	C^S	C^H	C^I	C^O
0.6	0.5	0.180	0.12	0.380	0.317	0.317	0.317
0.6	1.0	0.180	0.24	0.418	0.328	0.328	0.328
0.6	1.5	0.180	0.36	0.456	0.335	0.335	0.335
0.6	2.0	0.180	0.48	0.493	0.343	0.343	0.343
1.0	0.5	0.300	0.20	0.971	0.726	0.726	0.726
1.0	1.0	0.300	0.40	1.143	0.792	0.792	0.792
1.0	1.5	0.300	0.60	1.314	0.841	0.841	0.841
1.0	2.0	0.300	0.80	1.486	0.889	0.890	0.889
1.5	0.5	0.450	0.30	3.382	2.815	2.186	2.185
1.5	1.0	0.450	0.60	4.364	2.617	2.617	2.617
1.5	1.5	0.450	0.90	5.345	2.996	2.996	2.996
1.5	2.0	0.450	1.20	6.327	3.366	3.370	3.365

Table 4.1: Comparative cost performance of competitor policies for problems with $\mu_1 = 1,\ \mu_2 = 1$.

$\bar{\rho}$	ν	λ_d	λ	C^S	C^H	C^I	C^O
0.6	0.5	0.200	0.12	0.376	0.312	0.312	0.312
0.6	1.0	0.200	0.24	0.409	0.328	0.322	0.322
0.6	1.5	0.200	0.36	0.441	0.335	0.329	0.329
0.6	2.0	0.200	0.48	0.474	0.341	0.336	0.335
1.0	0.5	0.333	0.20	0.951	0.711	0.712	0.711
1.0	1.0	0.333	0.40	1.102	0.781	0.772	0.768
1.0	1.5	0.333	0.60	1.251	0.828	0.818	0.815
1.0	2.0	0.333	0.80	1.399	0.871	0.861	0.858
1.5	0.5	0.500	0.30	3.273	2.125	2.125	2.124
1.5	1.0	0.500	0.60	4.143	2.534	2.525	2.503
1.5	1.5	0.500	0.90	5.008	2.881	2.864	2.849
1.5	2.0	0.500	1.20	5.869	3.213	3.196	3.176

Table 4.2: Comparative cost performance of competitor policies for problems with $\mu_1 = 1,\ \mu_2 = 1.25$.

$\bar{\rho}$	ν	λ_d	λ	C^S	C^H	C^I	C^O
0.6	0.5	0.216	0.12	0.372	0.309	0.309	0.309
0.6	1.0	0.216	0.24	0.400	0.328	0.318	0.318
0.6	1.5	0.216	0.36	0.428	0.335	0.324	0.324
0.6	2.0	0.216	0.48	0.455	0.341	0.330	0.330
1.0	0.5	0.360	0.20	0.933	0.701	0.701	0.701
1.0	1.0	0.360	0.40	1.065	0.775	0.758	0.751
1.0	1.5	0.360	0.60	1.193	0.817	0.798	0.792
1.0	2.0	0.360	0.80	1.318	0.856	0.836	0.830
1.5	0.5	0.540	0.30	3.185	2.087	2.087	2.079
1.5	1.0	0.540	0.60	3.963	2.476	2.460	2.415
1.5	1.5	0.540	0.90	4.724	2.796	2.762	2.719
1.5	2.0	0.540	1.20	5.475	3.094	3.064	3.006

Table 4.3: Comparative cost performance of competitor policies for problems with $\mu_1 = 1,\ \mu_2 = 1.5$.

4.5 Numerical Study

We now present the results of computational investigation of the performance of the index policy developed in this chapter. We consider the closeness to optimality of the index policy for a range of scenarios sufficiently simple to allow the computation of a fully dynamic optimal routing policy via DP but sufficiently rich to yield insight. In addition to these heuristics we would expect the dynamic routing heuristic of Chapter 3 developed via the application of a single policy improvement step to some optimal static policy (for the overall system) to perform well in the problems considered here. We shall include results for these routing policies for comparative purposes. The reader should note that the development of the dynamic heuristics is substantially motivated by the inability of DP methods to produce solutions to problems of reasonable size. The earlier consideration of the choice of scenario can only emphasise this point. The problems studied all involve two stations each admitting one dedicated class in addition to the generic traffic. To keep notation simple, we write μ_1 for the service rate of the dedicated traffic at station 1, μ_2 for the dedicated service rate at station 2 and ν for the common service rate (at both stations) for the generic traffic. In all cases studied we assume a common arrival rate ($\lambda_1 = \lambda_2 = \lambda_d$, say) for the dedicated traffic. We consider three

scenarios, $(\mu_1 = 1, \mu_2 = 1)$, $(\mu_1 = 1, \mu_2 = 1.25)$ and $(\mu_1 = 1, \mu_2 = 1.5)$, which reflect an increasing dissimilarity between the stations through the exponential service times of the jobs dedicated to station m. Within each scenario a range of values for the remaining parameters are considered, with the *system traffic intensity*

$$\bar{\rho} = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} + \frac{\lambda}{\nu}$$

taking the values of 0.6 (light traffic), 1.0 (moderate traffic) and 1.5 (moderate to heavy traffic). For all chosen values of the system parameters, we computed C^S, C^{II}, C^I and C^O, namely, the long run average cost rate in (4.2) for the optimal static policy (of the overall system), the dynamic routing heuristic, the index policy and the optimal dynamic policy respectively. A simple formula is available for the calculation of C^S. All other costs were computed using DP value iteration, with the maximum queue lengths adjusted upward to achieve stability of the associated costs. Note that for some problems with $\bar{\rho} = 1.5$, the study required solutions to DP's with state space of around 3×10^6.

In Tables 4.1-4.3 we present the results of the numerical study. The dynamic policies enjoy an increasing cost advantage over the static policy as ρ grows. H and I enjoy a strong overall performance, however, I improves upon H as the stations become more dissimilar. In none of the above cases does C^I exceed C^O by more than 2% and in half the cases the difference is less than 0.1%. This evidence suggests that the proposed index heuristic is both simple and effective. We would expect its performance to be particularly strong in set ups where the spare capacity available to process generic traffic does not vary greatly between stations. The performance of I is especially appealing in comparison to H with regard to the computational effort required in determination of the required indices.

4.6 Conclusions

We described an approach to the development of index policies utilising the methodologies related to the class of restless bandit problems. By taking an approximative approach to Whittle's proposal for indexability we obtained a closed form station index that was shown to be an increasing and non-linear function of the workload at the station. A numerical study testifies to the strong performance of the routing policies.

The thesis has highlighted the practical importance of studying routing problems in a multi-class context in which traffic types may have widely differing cost and stochastic characteristics. However, such problems present a significant challenge for analysis. We described approaches to the development of routing policies on the basis of the level of information available to the system controller. The major achievements of our work has been the development of two heuristic dynamic routing policies for our complex multi-class routing problems. These are shown to have very strong performance and in many cases are close to optimal. There is scope for extending these analyses to cases in which the stations are modelled as a $M/G/1$ queueing system operating a nonpreemptive priority policy. In a full information routing model the system controller will require knowledge of the expired processing time of any job currently in service in addition to the vector of queue lengths at each station. Preliminary studies indicate that versions of both heuristic routing polices are available and are expected to perform well.

Given the strong performance of the routing heuristics, it would be very interesting to see how they perform in a wider routing context. In our models arriving jobs (are assumed to) patiently await service while incurring holding costs and the service stations are always operational. In 'real' problems this may not necessarily be the case. Jobs awaiting service could leave the system of their own accord, possibly to seek more efficient service elsewhere, or they could have have some (unknown) expiry date at which it is no longer necessary for it to be processed. Service stations themselves may breakdown and will be unable to process jobs until it can be repaired or may simply go on vacation for an undetermined amount of time. Such additions to our models add further complexities

to an already challenging problem. However, for suitably defined models incorporating information of customer loss rates or station breakdowns and their (random) duration, it could be possible to develop dynamic routing strategies by following similar approaches.

Bibliography

Altman, E. (2000), 'Applications of Markov Decision Processes in Communication Networks: a Survey', Rapport de Recherche 3984, INRIA.

Ansell, P. S., Glazebrook, K. D. & Kirkbride, C. (2003), 'Generalised 'join the shortest queue' policies for the dynamic routing of jobs to multi-class queues', *Journal of the Operational Research Society,* **54**, 371–378.

Ansell, P. S., Glazebrook, K. D., Niño-Mora, J. & O'Keeffe, M. (2003), 'Whittle's index policy for a multi-class queueing system with convex holding costs', *Mathematical Methods of Operations Research,* **57**, 21–39.

Atkins, D. & Chen, H. (1995), 'Performance evaluation of scheduling control of queueing networks: fluid model heuristics', *Queueing Systems,* **21**, 391–413.

Becker, K. J., Gaver, D. P., Glazebrook, K. D., Jacobs, P. A. & Lawphongpanich, S. (2000), 'Allocation of tasks to specialized processors: a planning approach', *European Journal of Operational Research,* **126**, 80–88.

Bellman, R. (1957), *Dynamic Programming*, Princeton University Press, Princeton.

Bertsimas, D. & Niño Mora, J. (1996), 'Conservation laws, extended polymatroids and multi-armed bandit problems: a polyhedral approach to indexable systems', *Mathematics of Operations Research,* **21**, 257–306.

Braun, T. D., Siegel, H. J. & Maciejewski, A. A. (2001), Heterogeneous Computing: Goals, Methods and Open Problems, *in* 'Proceedings of the 8th International Conference on High Performance Computing', pp. 307–320.

Brooks, S. P. & Morgan, B. J. T. (1995), 'Optimization using simulated annealing', *Statistician*, 44, 241–257.

Chang, C. S. (1992), 'A new ordering for stochastic majorization: theory and applications', *Advances in Applied Probability*, **24**, 604–634.

Coffman, E. & Mitrani, I. (1980), 'A characterization of waiting time performance realizable by single server queues', *Operations Research*, **28**, 810–821.

Cox, D. R. & Smith, W. L. (1961), *Queues*, Methuen, London.

Dacre, M. J. (1999), Stochastic scheduling in networks, PhD thesis, Newcastle University.

Dacre, M. J. & Glazebrook, K. D. (2002), 'The dependence of optimal returns from multi-class queueing systems on their customer base', *Queueing Systems*, 40, 93–115.

Dacre, M. J., Glazebrook, K. D. & Niño-Mora, J. (1999), 'The achievable region approach to the optimal control of stochastic systems (with discussion)', *Journal of the Royal Statistical Society*, **B61**, 747–791.

Edmonds, J. (1970), Submodular functions, matroids and certain polyhedra, *in* 'Proceedings of Calgary International Conference on Combinatorial Structures and their Applications', Gordon and Breach, New York, pp. 69–87.

Ephremides, A., Varaiya, P. & Walrand, J. (1980), 'A simple dynamic routing problem', *IEEE Transactions on Automatic Control*, **AC-25**, 690–693.

Foster, I. & Kesselman, C., eds (1998), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufman, San Francisco.

Garbe, R. & Glazebrook, K. D. (1998), 'Submodular returns and greedy heuristics for queueing scheduling problems', *Operations Research*, **36**, 336–346.

Gelenbe, E. & Mitrani, I. (1980), *Analysis and Synthesis of Computer Systems*, Academic Press, London.

Gelenbe, E. & Pekergin, F. (1993), Load balancing pragmatics, Technical report, EHEI, Université René Descartes.

Gittins, J. C. (1979), 'Bandit processes and dynamic allocation indices (with discussion)', *Journal of the Royal Statistical Society,* **B41**, 141–177.

Glazebrook, K. D. & Kirkbride, C. (2004), 'Index policies for the routing of background jobs', *Naval Research Logistics,* **51**, 856–872.

Glazebrook, K. D. & Mitchell, H. (2002), 'An index policy for a stochastic scheduling model with improving/deteriorating jobs', *Naval Research Logistics,* **49**, 706–721.

Glazebrook, K. D., Niño-Mora, J. & Ansell, P. S. (2002), 'Index policies for a class of discounted restless bandits', *Advances in Applied Probability,* **34**, 754–774.

Harrison, J. M. & Wein, L. (1989), 'Scheduling networks of queues: heavy traffic analysis of a simple open network', *Queueing Systems,* **5**, 265–280.

Hordijk, A. & Koole, G. (1990), 'On the optimality of the generalized shortest queue policy', *Probability in the Engineering and Informational Sciences,* **4**, 477–487.

Kleinrock, L. (1976), *Queueing Systems,* Vol. II: Computer Applications, Wiley, New York.

Kleinrock, L. (2002), 'Creating a mathematical theory of computer networks', *Operations Research,* **50**, 125–131.

Klimov, G. P. (1974), 'Time sharing systems I', *Theory of Probability and its Applications* **19**, 532–551.

Koole, G. (1996), 'On the pathwise optimal Bernoulli routing policy for homogeneous parallel servers', *Mathematics of Operations Research,* **21**, 469–476.

Krishan, K. R. (1987), Joining the right queue: a Markov decision rule, *in* 'Proceedings of the 28th IEEE Conference on Decision and Control', pp. 1863–1868.

Liu, Z. & Righter, R. (1998), 'Optimal load balancing on distributed homogeneous unreliable processors', *Operations Research,* **46**, 563–573.

Liu, Z. & Towsley, D. (1994), 'Optimality of the round-robin policy', *Journal of Applied Probability,* **31**, 466–478.

Niño-Mora, J. (2001), 'Restless bandits, partial conservation laws and indexability', *Advances in Applied Probability,* **33**, 76–98.

Niño-Mora, J. (2002), 'Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach', *Mathematical Programming,* **93**, 361–413.

Papadimitriou, C. H. & Tsitsiklis, J. N. (1999), 'The complexity of optimal queueing network control', *Mathematics of Operations Research,* **24**, 293–305.

Puterman, M. L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York.

Ross, K. W. & Yao, D. D. (1991), 'Optimal load balancing and scheduling in a distributed computer system', *Journal of the Association for Computing Machinery,* **38**, 676–690.

Shanthikumar, J. G. & Yao, D. D. (1992), 'Multi-class queueing systems: polymatroidal structure and optimal scheduling control', *Operations Research,* **40**, 293–299.

Tantawi, A. N. & Towsley, D. (1985), 'Optimal static load balancing in distributed computer systems', *Journal of the Association for Computing Machinery,* **32**, 445–465.

Tjims, H. C. (1994), *Stochastic Models: an Algorithmic Approach*, Wiley, Chichester.

Tsoucas, P. (1991), The region of achievable performance in a model of Klimov, Technical report, IBM T.J.Watson Research Center, Yorktown Heights, NY.

Weber, R. R. (1978), 'On the optimal assignment of customers to parallel queues', *Journal of Applied Probability,* **15**, 406–413.

Weber, R. R. (1980), 'On the marginal benefit of adding servers to G/GI/m queues', *Management Science,* **26**, 946–951.

Weber, R. R. & Weiss, G. (1990), 'On an index policy for restless bandits', *Journal of Applied Probability,* **27**, 637–648.

Weber, R. R. & Weiss, G. (1991), 'Addendum to "On an index policy for restless bandits"', *Advances in Applied Probability* **23**, 429–430.

Whittle, P. (1988), 'Restless bandits: activity allocation in a changing world', *Journal of Applied Probability,* **A25**, 287–298.

Whittle, P. (1996), *Optimal Control: Basics and Beyond*, Wiley, New York.

Winston, W. (1977), 'Optimality of the shortest line discipline', *Journal of Applied Probability,* **14**, 181–189.